

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

PC

特集 XCのための傾向と対策

THE SOFTOUCH SPECIAL II 特大GAME REVIEW

シュミレーションプログラミング入門

特別企画アナログジョイスティックの製作

12
1990

SOFT
BANK

オーノエックス
定価560円



X68000を象徴する

高解像度自然色グラフィック

X68000の数々の注目すべき能力のうち、とりわけ際立っているのが「強力なグラフィック能力」だといえましょう。流通する多くのコンピュータが8色表示という環境の中で、65,536色同時表示(512×512ドットモード時)というハイスぺックで登場。発売後、数年経過した現在でも初代X68000、CZ-600Cカタログの表現が、そのまま通用します。「クロームやチタニウムに代表される高品位な金属の質感、金、銀表現、人の目に映る色や形状をほとんどありのままに表現し得る自然色グラフィックスが、これまでのCGイメージを一刷新します。」……まさにX68000を象徴するこのスペックは、ごく最近までパーソナルコンピュータでは不可能とされた“レイトレーシング”にしても、フレームバッファなどの増設なしで実現可能とし、その設計思想の先見性と、コストパフォーマンスの高さを雄弁に物語っています。



65,536色C.G. ▲

もう一步ふみこんだ

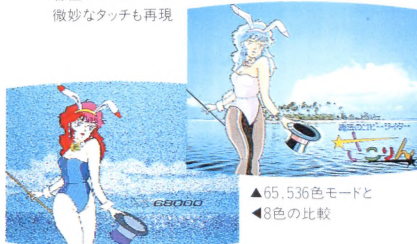
オリジナルアートの世界を

65,536色を駆使するには、カラーイメージユニット(CZ-6VT1)やカラーイメージスキャナ(CZ-8NS1)などの画像入力機器が役立ちます。これらの機器は写真、水彩画、油絵、色鉛筆やパステルなど、ビジュアル化

のあらゆるアプローチをディスプレイ上にシミュレートし、それぞれの特徴を活かしながら、独自のCGの世界を創造します。まさにAVに強いと言われるX68000の真骨頂を示すものといえましょう。また一般に色数が少ないとされるアニメ調の絵にしても、肌色などの中間色に対する妥協を許しません。



各種ペイントの ▲
微妙なタッチも再現



▲65,536色モードと
▲8色の比較

クリエイティブワークを広げる

先駆の独立3画面設計

そしてX68000のメモリアーキテクチャは、68000CPUのもつ広大なアドレス空間を活かして、テキスト、グラフィック、スプライトの3画面を独立構造として装備した独自の画面設計です。文字、CG、キャラクタをプライオリティつきで重ね合わせ表示する、これまでむずかしかったビジュアル表現も造作なくこなすハイアビリティがクリエイティブワークの幅を広げます。なかでもゲームデザインを強力にサポート、アニメーションと呼ぶにふさわしい興奮のシーンを展開できるスプライトとバックグラウンドはクリエイターの血を騒がせています。またパソコンテレビX1の血統を受け継いだスーパーインポーズ機能、テレビやビデオの映像をバックに通信したりプログラミングにチャレンジしたり……こうした環境を標準で楽しめるところにも、X68000の



▲65,536画像取り込みを迫力の21型ディスプレイで……

シャープX68000パソコン教室開催中
●会場：市ヶ谷教室 シャープ東京支社ビル
●コース：入門コース・表集計コース・音楽コース・絵画コース
●申込受付電話番号：(03) 260-8365
●受講料：2,000円(税別)

68買ったら
EXE
クラブ
に入ろう!

本体同梱の入会申込ハガキを送るだけで、無料入会。

- ⑤ **メリット1**：会員No.入り、オリジナル**会員証電卓**ももらえる。
 - ⑤ **メリット2**：各種フェアご優待・イベントご案内等、数々の特典アリ。
 - ⑤ **メリット3**：10月1日スタート！X68000の活用情報が手に入る
- 「EXEおみこし活動」に参加できる!!

ステップアップサービス(有料)
「おみこしかつぎ人」制度も新設

ますます楽しい
X68000EXEクラブ!

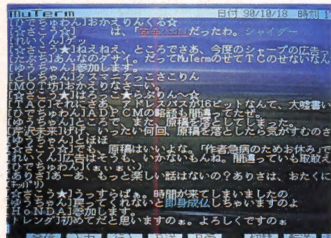
詳細はX68000販売店店頭で
ポスター・おみこしPressをご覧ください。

実証する 意味。

敢えてX68000のグラフィックアビリティを



能力の一端をかいまみることができます。



▲映像をバックに通信も

フレキシブルな

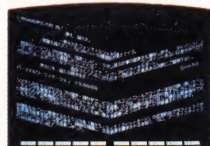
ビットマップ方式のテキストRAM

もちろん512KバイトVRAMを搭載したテキスト機能も、その特色の一つに数えられます。一般に使用される80桁表示をこえた96桁のテキスト表示(ANK)もさることながら、ビットマップ方式の採用により、各キャラクタのみでなく1ドット毎の細かい制御が可能(MELT.R)。加えてユーザー独自の字体をデザインして使用することも比較的容易です。また、このテキストVRAMはグラフィックの表示も可能であり、SX-WINDOWのPIX形式のファイルやPDSのMAKI形式で利用されており、他機種の8/16色グラフィックとのデータの共有を実現しています。このテキストVRAMを利用してオリジナルのTITLE.SYSをつくったりスーパーインポーズ機能を

活かしてTVパソコンとして楽しんだり、カスタマイズの世界がどんどん広がっていきます。2、3のアプリケーションで納得するパソコン、それはそれとして、ここには文房具としてでなく創造するものとしてのパソコンの世界があります。

68000CPUとX68000

の可能性は、まだほんの頭を覗かせただけかもしれません。



テキストのドットコントロール(MELT.R)▲



ユーザーフォント作成例▲



オリジナルTITLE.SYS▶

レイトレーシング: 立体感のあるC.G.を描画するため、目から光源へさかのぼって反射鏡をシミュレートする技法。光線追跡法という。かつてはその計算に数日を要したが、今では高速演算プロセッサとの組み合わせでスピーディに処理でき、パソコンレベルでもチャレンジ可能。**フレームバッファ:** 画像データを記憶するメモリ(グラフィックRAM)、フレームメモリという。通常、パソコンの拡張スロットに差し込んで用い、高度なCGを実現できるが、高価なのでちょっと手が出づらい。

中間色: 肌色や金属色など、8色や16色表示では表現しにくい色調。

広大なアドレス空間: 多くの機種では、CPU(中央制御装置)が巨大なメモリを扱ってもOS(オペレーティングシステム)の関係で使えるメモリが限られるという矛盾した状態になる。もちろんX68000ではそのようなことはない。

テキスト: 文字や記号だけを表示する画面。通常は16

×16ドットのキャラクタを1文字単位でコントロールする。**グラフィック:** 絵を書くための画面。1ドット毎にコントロールでき、ドット(点)の集合として描画する。

スプライト: グラフィック画面上で重ね合わせをする機能。ゲームのキャラクタなどを高速で動かすための機能のひとつ。X68000では、水平32スプライト、1画面128スプライト同時表示というハイスペックを実現している。

VRAM: プライムあるいはビデオRAM。CRTディスプレイに表示するためのメモリ。RAMの内容に応じて、文字やグラフィックが表示されるように処理されたメモリ回路。**ANK:** Alphabet, Numeric, Kana. アンクと読む。アルファベット、数字、かなの3種類の半角文字をANK文字と呼んでいる。

MELT.R: On氏作のPDS。オリジナルアイデアは他機種のもの。参考までに、このプログラムでは、文字の各ドットは操作しておらず、表示位置をドット単位で操作する。**ユーザー独自の字体:** 丸文字、変体少女文字に始まって、最近ではロシア語に至るまで、様々な字体が広がっている。従来の「外字」は、個人のものであったが、通信の広がりにより同じフォントを使うといった事態も生じている。

MAKI形式: 草の根ネット「まきちゃんNET」において開発された640×400ドットアナログ16色画像のデータファイル用フォーマット。X1/turbo、MZ25/28など多くの機種間で画像ファイル共有の架け橋となる。

TITLE.SYS: 通常は、電源を入れたら「SAKO-DOS V2.1 Copyright1990 SAKORIN」などといった渴いたメッセージが表示されるが、X68000の場合は、画像がこれのかわりを勤める。これが「TITLE.SYS」であり、知識があれば自分だけのTITLE.SYSも作れる。

【今回の広告に使用したツール一覧】SCANTOOL.X(シャープ製)、MONO.X/GtoT8.X/TITLE.X/GTOCONV.X/TXCLR.R(WOODY RINN氏作)、MFGED.X/MFLOAD.X(結城 見氏作)、GL3LD.X/(HoNDA氏作)、ACF.X(夢職人氏作)、MuTerm.X(はちくん氏作)、PIC.R(柳沢 明氏作)、MELT.R(On氏作)

ゴメンナサイ!! ●ADPCMの英文中、「Code」の文字が欠落しておりました。また●「アドレスバスは16ビットながら……」などと大変な間違いをしておりました。広島県の本様は多数の皆様から指摘を受け、日々反省しております。正しくは「データバス……」でした。

SUPER HD
本体+キーボード+マウス+トラックボール
CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

EXPERT II
本体+キーボード+マウス+トラックボール
CZ-603C-BK(ブラック)--GY(グレー) 標準価格338,000円(税別)
HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

PRO II
本体+キーボード+マウス
CZ-653C-BK(ブラック)--GY(グレー) 標準価格285,000円(税別)
HDタイプ CZ-663C-BK(ブラック)--GY(グレー) 標準価格395,000円(税別)



充実のディスプレイ ラインアップ

※印の商品は在庫僅少です。

15型カラーディスプレイテレビ(ドットピッチ0.39mm) CZ-602D-BK(ブラック)--GY(グレー)
標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm) CZ-605D-BK(ブラック)--GY(グレー)
標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm) CZ-613D-TN(チタンブラック)--BK(ブラック)--GY(グレー)
標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)

14型カラーディスプレイ(ドットピッチ0.31mm) CZ-603D-BK(ブラック)--GY(グレー)
標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm) CZ-604D-BK(ブラック)--GY(グレー)
標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)
21型カラーディスプレイ(ドットピッチ0.52mm) CU-21HD-BK(ブラック)
標準価格148,000円(スピーカー2個同梱・税別)

X68000 自分流カード デザインコンペ 作品大募集

〈応募要領〉●応募方法/X68000で作成したポストカードサイズのデザインカードを送って下さい。(ソフトは自由)●作品分類/部門A: クリスマスカード、ニューイヤーカード 部門B: バレンタインカード、バースデイカード 部門C: 暑中見舞カード・サークル・趣味の会お知らせカード●賞/A・B・C各部門毎に優秀作品を選考、オリジナルカレンダーに掲載してプレゼントします。※優秀作品賞: 掲載作品応募者に、カレンダー及びオリジナル表彰額を進呈。※参加賞: 応募者全員に、カレンダーを進呈。(応募作品に関わる諸権利は主催者に帰属するものとして作品の返却はいたしません)
●応募期間/1990年10月1日～1991年2月28日(消印有効)

詳細はX68000販売店店頭で、
チラシ・ポスターをご覧ください。

本広告に関するご意見をお寄せください。下記大阪本社宣伝部「☆さこう★係」まで

●お問い合わせは…

シャープ株式会社

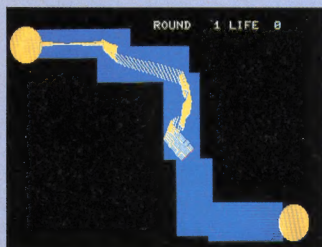
シャープ(株)電子機器事業本部システム機器営業部
〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部
〒162 東京都新宿区西八幡町8番地 ☎(03)260-1161(大代表)



特集 XCのための傾向と対策



XBASToC CHECKER



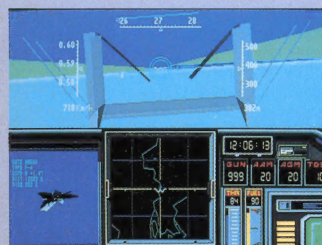
(て)のショートプロはーい



イメージファイト



NAIOUS



エア-コンバット (遊撃王II)

コン

C O N T

●特集

105 XCのための傾向と対策

- | | | |
|-----|--|------|
| 106 | Cコンパイラのアウトライン
XC ver.2.0 ガイドマップ | 荻窪 圭 |
| 112 | XCを支援するおいしいツール
ソースコードデバッガを使ってみよう | 泉 大介 |
| 116 | 貴方のプログラミングを支援する
縁の下のプリプロセッサ | 中森 章 |
| 121 | 基礎知識からプログラミングへ
Cライブラリ活用の手引き | 丹 明彦 |
| 127 | 多数のソースファイルを管理する
XCにMAKEが付いてきた | 中森 章 |
| 129 | MIDI制御が加わった
新しい音楽ドライバOPMDRV2.X | 西川善司 |
| 131 | BASICコンパイラ派に贈る便利ソフト
XBASToC CHECKER | 西川善司 |

●Orx 3周年特別企画

- | | | |
|----|-----------------|------|
| 40 | 愛読者プレゼント | |
| 86 | アナログジョイスティックの製作 | 石上達也 |

●THE SOFTOUCH

- | | | |
|----|-----------------------------------|------|
| 42 | SOFTWARE INFORMATION
話題のソフトウェア | |
| 46 | THE SOFTOUCH SPECIAL
イメージファイト | 中野修一 |
| 48 | ジェミニウイング | 山田純二 |
| 50 | NAIOUS | 影山裕昭 |
| 52 | FZ戦記 アクシス | 斎藤 晋 |
| 54 | 機甲師団 | 山田純二 |
| 56 | ニューラル・ギア | 吉田賢司 |
| 57 | 闇の血族 完結編 | 古村 聡 |
| 58 | 熱血高校ドッジボール部サッカー編 | 荻窪 圭 |
| 60 | エア-コンバット (遊撃王II) | 西川善司 |
| 62 | バルーサの復讐 | 浦川博之 |

〈スタッフ〉

●編集長/前田 徹 ●副編集長/植木章夫 ●編集/岡崎栄子 浅井研二 ●協力/有田隆也 中森 章
後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田
敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ/杉山和美
●イラスト/永沢しげる 山田晴久 小栗由香 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子
AD GREEN ●校正/グループごじら



表紙絵：須藤 牧人

EN TS

●カラー紹介

38 SHOW REPORT
エレクトロニクスショウ&データショウ'90

39 THE USER'S WORKS
グラディウスX1turbo

●シリーズ全機種共通システム

133 THE SENTINEL

134 STACKコンパイラ

平井真二

●読みもの

166 猫とコンピュータ 第54回
マニアの砦にて

高沢恭子

169 X-OVER NIGHT 第7話
スキーは豪華に?

高原秀己

●連載/紹介/講座/プログラム

63 大人のためのX68000 第3回
スプレッドシートの精神

荻窪 圭

67 シミュレーションプログラミング入門 第1回
コンピュータシミュレーションの世界

華門真人

73 (で)のショートプロバてい その15
テクニックは偉大なのだ!

古村 聡

78 Oh! X LIVE in '90
グラディウスIIIより Sand Storm (X68000)
メタルサイトより Into The Shadow (X1/turbo)

柏木勝利

高橋哲史

82 ハードウェア工作入門 (6)
A/Dコンバータ その3

三沢和彦

92 X-BASICプログラミング調理実習 (16)
カード型データベース(2)

泉 大介

97 X68000マシン語プログラミング Chapter_13H
C, X-BASICの関数を作成する

村田敏幸

142 ようこそここへC言語 [第3回]
制御構造って何だろう

中森 章

148 マシン語カクテル in Z80's Bar 第17回
私はエディタ, 原稿まだかな

山田純二

153 X1/turbo用ディスク管理プログラムINTEGRAL X1
メニューによるファイル管理

亀田雅彦

164 X68000 CARD, FNC用カードゲーム
COUPLE

青木実千男

Oh!X INDEX '90.....170

ペンギン情報コーナー.....174

FILES Oh!X.....176

Oh!X質問箱.....178

STUDIO X.....180

編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey.....184

1990 DEC. 12

UNIXはAT&T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86, CP/M-68K, CP/M-8000, DR-DOSはDIGITAL RESEARCH
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACROS, MS CはMICRO
SOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事會
WordStar, WordMasterはWORDSTAR International
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTER
NATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他、プログラム名、CPUは一般に各
メーカーの登録商標です。本文中では"TM", "R"マー
クは明記していません。
本誌に掲載されたプログラムの著作権はプログラム
作成者に保留されています。著作権上、PDSと明記さ
れたもの以外、個人で使用するほかの無断複製は禁
じられています。

■広告目次

アイビット電子	194・195
アイレム	27
アクセス	200
アートディンク	10
AVCフタバ電機	188
エムエーシー ハミングバードソフト	12
エルフ	28
オーエーブレイン	196
オーエーランド	29
キャスト	9
計測技研	190・191
工画堂スタジオ	22・23
コナミ	14・17
ザインソフト	11
サン・ミュージカル・サービス ...	199(上)
J & P	表3
システムサコム	24・25
シャープ	表2・表4・1・4-7
ソフトクリエイト	198
ソフマップ	192・193
九十九電機	30・31
ティーアンドイーソフト	13
DISKシャトル高槻	197
デンキヤ	189
日本ソフテック	8
パソコンプラザオクト	34・35
P & A	32・33
ビクター音楽産業	20・21
ヘルツ	26
ボーステック	18・19
満開製作所	187



ディスプレイ関連

カラーディスプレイテレビ



15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

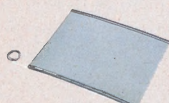


15型カラーディスプレイテレビ
CZ-605D-BK-GY
標準価格 115,000円(税別)
(スピーカー2個・チルトスタンド同梱)



15型カラーディスプレイテレビ
CZ-613D-TN-BK-GY
標準価格 135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

CRTフィルター



高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14/15型用)

カラーディスプレイ



14型カラーディスプレイ
CZ-603D-BK-GY
標準価格 84,800円(税別)
(チルトスタンド同梱)



14型カラーディスプレイ
CZ-604D-BK-GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)



21型カラーディスプレイ
CU-21HD
標準価格 148,000円(税別)
(スピーカー2個同梱)

チューナー



RGBシステムチューナー
CZ-6TU-BK-GY
標準価格 33,100円(税別)
(リモコン付)

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスカナ用パラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。
※2 CZ-603D 604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。 ※3 ビデオ出力は15.75kHzテレビ標準信号です。また、拡張I/Oスロットは2スロット使用します。
※4 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。 ※5 CZ-600C、601C、602C、603C、611C、612C、613C、652C、653C、662C、663Cに使用の場合は、別売のSCSIボード(CZ-6BS1)が必要です。(但し、CZ-623Cは不要)
また、X68000用OS Human68K ver.2.0以上にてご使用ください。(光磁気ディスクカートリッジは別売のJY-701MPA標準価格30,000円(税別)をご使用ください。) ※6 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格

アートツール

画像入力



カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円(税別)



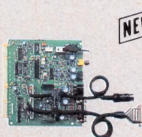
スカナ用パラレルボード
CZ-6BN1
標準価格 29,800円(税別)

映像入力



カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

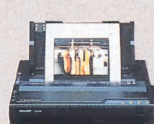
映像出力



ビデオボード※3
CZ-6BV1
標準価格 21,000円(税別)

プリンタ

カラープリンタ



24ドット
熱転写カラー漢字プリンタ
★CZ-8PC3
標準価格 65,800円(税別)
(信号ケーブル同梱)



48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)

カラービデオプリンタ



カラービデオプリンタ
CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット



カラーイメージジェット※4
IO-735X
標準価格 248,000円(税別)
(信号ケーブル別売)

ドットプリンタ



24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)



24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)



24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

ファイル

光磁気ディスク



光磁気ディスクユニット※5
(594MB)
CZ-6MO1
標準価格 450,000円(税別)
(SCSIケーブル同梱)
※光磁気ディスクカートリッジは別売です。別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。

ハードディスク



ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)



増設用ハードディスク
ドライブ(40MB)
(CZ-602C/603C/652C/653Cに内蔵)
CZ-64H
標準価格 120,000円(税別)
(取付費別)

※取付に関してはシャープお客様ご相談窓口にてご相談ください。

W-turbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ

●21型カラーディスプレイ※1 CU-21HD 148,000円

映像・画像入力編集装置

●カラーイメージスキャナ CZ-8NS1 188,000円
●カラーイメージボードII CZ-8BV2 39,800円

●立体映像セット ★CZ-8BR1 29,800円
●パーソナルテロップ※2 CZ-8DT2 44,800円

FM音源

●ステレオタイプFM音源ボード CZ-8BS1 23,800円
スピーカー(2本1組)標準装備、ミュージックツール同梱

プリンタ

●24ピンカラー漢字プリンタ(80桁) CZ-8PG1 130,000円
●24ピンカラー漢字プリンタ(136桁) CZ-8PG2 160,000円

●24ピン漢字プリンタ(136桁) CZ-8PK10 97,800円
●24ドット熱転写カラー漢字プリンタ ★CZ-8PC3 65,800円
●48ドット熱転写カラー漢字プリンタ CZ-8PC4 99,800円
●48ドット熱転写カラー漢字プリンタ CZ-8PC4-GY 99,800円
●カラービデオプリンタ CZ-6PV1 198,000円
●カラーイメージジェット IO-735X 248,000円

ファイル

●ミニフロッピーディスクユニット(2HD・2D)※3 ★CZ-520F 118,000円

お望みのパワーシステムへ。



シャープペリフェラルファミリー
68000



ボード

拡張メモリ



1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)



1MB増設RAMボード
(CZ-601C/611C/652C/
653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)



2MB増設RAMボード*6
CZ-6BE2
標準価格 79,800円(税別)



4MB増設RAMボード*6
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス



ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)



GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)



増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)



SCSIボード*7
CZ-6BS1
標準価格 29,800円(税別)
(ソフトウェア(SCSIユーティリティ)同梱)

数値演算プロセッサ



数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX



FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI



MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム



モデムユニット*8
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル



RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)



RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード



LANボード
CZ-6BL1
標準価格 268,000円(税別)
(イーサネット用)
CZ-6BL2
標準価格 298,000円(税別)
(イーサネット/ターボネット両用)
*電源ユニット・ソフトウェア
(ネットワークライブラリVer.1.0)同梱

入力



インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)



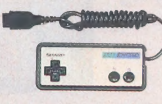
マウス・トラックボール
CZ-8NM3
標準価格 9,800円(税別)



トラックボール
CZ-8NT1
標準価格 13,800円(税別)



マウス
CZ-8NM2A
標準価格 6,800円(税別)



ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット



拡張I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/
611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー



アンプ内蔵
スピーカシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック



システムラック
(CZ-600C/601C/602C/603C/
611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。*7 CZ-600C、601C、602C、603C、611C、612C、613Cに装着の場合、I/Oスロット2に装着ください。
CZ-652C、653C、662C、663Cに装着の場合はI/Oスロット4に装着ください。また、CZ-6BG1、6BU1、6BL1、6BL2、6BN1などのボードは、接続コネクタとの関係で本ボードとの併用はできませんのでご注意ください。なお、本ボードは68000用OS Human
68K ver.2.0以上にてご使用ください。*8 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)*4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード*5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス*6	CZ-8BF1	14,800円

●JIS第1水準漢字ROM*7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ*8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E-B	5,800円
●高性能CRTフィルター*9	BF-68PRO	19,800円
●スキャナ用パラレルボード*10	CZ-8BN1	27,800円

●品番中の-表示は、B(ブラック)・E(オフスグレイ)を示します。*1 X1ターボシリーズ用 *2 CZ-862Cには接続できません *3 X1ターボシリーズ用 *4 CZ-830C用 *5 X1シリーズ用 *6 CZ-850CでCZ-520Fを使用する場合に必要 *7 CZ-800C、801C、802C、803C、811C、820C用 *8 CZ-820C、822C、830C用 *9 I4/I5型用 *10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

ハイアビリティを実証する多彩なソフトウェア。

ドロー編集、WYSIWYG印刷、 こんなC.G.ツールが欲しかった。

本格的なロゴタイプやPOPを簡単に作成できるグラフィックツールです。優先順位が任意に指定できるドローセル、ペイントセル、テキストセルの3つの仮想セルで、目的にあった自由なグラフィックが駆使できます。また印刷は、画面イメージがそのまま印刷イメージとなるWYSIWYG(What You See Is What You Get)を実現。A6/A5/A4/A3/B6/B5/B4/葉書サイズで8色カラー印字できます。



〈ドローセル〉ベジェ曲線によって少ないデータ量でも複雑な絵を描くことができます。エンベロープ変形を始めとした豊富な編集機能を持っており、拡大、縮小しても絵の美しさは変わりません。またテキストセルで作成したベクトルフォントデータを自由に變形し、オリジナルロゴタイプやPOPを作成できます。

〈ペイントセル〉ペンやエアブラシ、ペンキなどを使って、ピクセルで構成されたビットマップ図形を描くことができます。また、「NEW PrintShop PRO-68K」や「X-BASIC」、「Z's STAFF PRO-68K」のデータ取り込みやイメージスキャナによる取り込みをサポートしています。

〈テキストセル〉通常の文字入力機能に加え、ベースライン変形などの多彩な編集機能によって自由に文字の加工ができます。また英数文字のベクトルフォントを標準装備。さらに「Z's STAFF PRO-68K Ver2.0」、「書体倶楽部」の日本語ベクトルフォントが利用可能。また、内蔵の漢字ROMフォントも自動的にベクトルフォントデータに変換しますので、簡単に日本語ロゴタイプを作成することができます。

※「Z's STAFF PRO-68K」、「書体倶楽部」は、㈱Zeitの製品です。
※本ソフトの動作には、メインメモリ2MBが必要です。

CANVAS PRO-68K CZ-249GS
標準価格29,800円(税別)

●主として個人用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

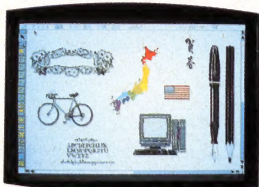
海のデータ/動物のデータ/スポーツのデータ/鳥のデータ/人物のデータ/食物のデータ/昆虫のデータ



CANVAS PRO-68K
ドローグラフィックライブラリ VOL.1
CZ-255GS 標準価格8,800円(税別)

●主としてビジネス用のさまざまなジャンルのデータが収められているドローグラフィックデータ集です。

OA関係のデータ/飾りのデータ/コンピュータ関連のデータ/POPのデータ/国旗のデータ/字体のデータ/地図のデータ/乗り物のデータ



CANVAS PRO-68K
ドローグラフィックライブラリ VOL.2
CZ-256GS 標準価格8,800円(税別)

バージョンアップされたCコンパイラ と、強力なBASTOCチェッカー。

ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Gのソースレベルでデバッグできる「ソースコードデバッガ」を搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(830種以上)となっています。G言語の標準であるANSI規格準拠をさらに強化。「プログラム保守ユーティリティ(MAKE)」や「ライブラリアン」など各種ツールを追加しました。その他「BASIC-Cコンバータ」、「アセンブラ」、「リンク」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「コンバータ」などのツールが装備されています。

※C compiler PRO-68K (CZ-211LS)を既にお持ちの方は、登録カードをもとに有償バージョンアップを行います。
※本ソフトの動作にはメインメモリ2MBが必要です。



COMPILER PRO-68K ver2.0 CZ-245LS
標準価格44,800円(税別)

トラブルエラーの悩み解消!

「XBASToC」の強力ツールの登場です。

X-BASICプログラムのコンパイル時、発見しづらいトラブルエラーに悩まされていたプログラムの問題点をひとつひとつ指摘。エラーとなる直接原因だけでなく、注意項目も指摘します。これにより、X-BASICでは実行できたのにコンパイルするとエラーが発生する、といったプログラムの修正が簡単にできます。

●指摘したトラブルの結果を、画面やプリンタなどの外部デバイスに簡単に出力できます。●エラーラインとエラーレポート、2つのエラーファイルを自動的に生成。●グラフィカルな画面による簡単操作。●コマンドラインからダイレクトに操作を指定。バッチファイルに組み込むなどの修正作業の自動化が可能。●GP-IBボード(CZ-6BG1)とユニバーサルI/Oボード(CZ-6BU1)付属の拡張外部関数もコンパイル可能。

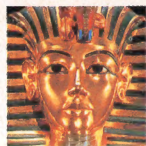


※X-BASICプログラムをコンパイルするためには、別売の「C compiler PRO-68K」(CZ-211LS)または「C compiler PRO-68K ver2.0」(CZ-245LS)が必要です。

XBASToC CHECKER PRO-68K
CZ-260LS 標準価格9,800円(税別)



お望みのワークベンチへ。



シャープオリジナルソフトウェア
68000

Hyperword

■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。



TOP給与計算エキスパート

■CZ-228BS 標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計

■CZ-227BS 標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CYBERNOTE PRO-60K

■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。



CARD PRO-60K

■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-60K用システム手帳リフィル集

■CZ-241BS 標準価格9,800円(税別)

CARD PRO-60K用活用フォーム集

■CZ-242BS 標準価格9,800円(税別)

Stationery PRO-60K

■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationeryPRO-60Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。



DATA PRO-60K

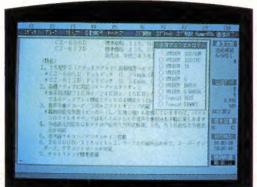
■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-60K

■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。



通信ツール

Communication PRO-60K ver.2.0

■CZ-257CS 標準価格19,800円(税別)

Communication PRO-60Kのバージョンアップ版です。MNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポートしています。

※バージョンアップ対応中。

NEW PrintShop PRO-60K

■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームブロッグディビリティツール。

グラフィックライブラリ VOL.1

■CZ-235GS 標準価格8,800円(税別)

グラフィックライブラリ VOL.2

■CZ-236GS 標準価格8,800円(税別)

SX-WINDOW ver.1.0

■CZ-259SS 標準価格6,800円(税別)

複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。IOCSコールを利用したソフトの処理速度を高速化するIOCS.Xを付属。



OS-9/X68000

■CZ-219SS 標準価格29,800円(税別)

マルチタスク機能、リアルタイム機能を活かした使いやすく機能的なOS環境を提供します。

※OS-9はマイクロウェア社の登録商標です。

Human68k ver.2.0

■CZ-244SS 標準価格9,800円(税別)

システムパフォーマンスをさらに高める処理機能を付加したHuman68kの最新バージョンです。

THE 福袋 V.2.0

■CZ-224LS 標準価格9,980円(税別)

AI-68K(Staff LISP/OPS PRO-68K)

■CZ-234LS 標準価格188,000円(税別)

サウンドツール

Musicstudio PRO-60K ver.1.1

■CZ-252MS 標準価格28,800円(税別)

MUSIC PRO-60K (MIDI)

■CZ-247MS 標準価格28,800円(税別)

ソングライブラリ<101曲集>

■CZ-248MS 標準価格8,800円(税別)

Sampling PRO-60K

■CZ-215MS 標準価格17,800円(税別)

SOUND PRO-60K

■CZ-214MS 標準価格15,800円(税別)

MUSIC PRO-60K

■CZ-213MS 標準価格18,800円(税別)

シューティングゲーム

<ツインビー>

■CZ-217AS 標準価格7,800円(税別)

© KONAMI. 1988

シューティングゲーム

<沙羅曼蛇>

■CZ-218AS 標準価格8,800円(税別)

© KONAMI. 1989

ブロックゲーム

<アルカノイド>

■CZ-222AS 標準価格7,800円(税別)

© TAITO CORP. 1987

ドライブゲーム

<フルスロットル>

■CZ-231AS 標準価格8,800円(税別)

© TAITO CORP. 1988

スポーツゲーム

<熱血高校ドッジボール部>

■CZ-232AS 標準価格7,800円(税別)

© TECHNOS JAPAN CORP. 1988

アクションゲーム

<バックマニア>

■CZ-233AS 標準価格7,800円(税別)

© NAMCO

アクションゲーム

<ニュージランドストーリー>

■CZ-230AS 標準価格8,800円(税別)

© TAITO CORP. 1989

スポーツゲーム

<V/BALL>

■CZ-246AS 標準価格7,900円(税別)

© TECHNOS JAPAN CORP. 1989

バイクレーシングゲーム

<スーパーハンクオン>

■CZ-238AS 標準価格8,800円(税別)

© SEGA 1987

ジェットヘリ・シミュレーションゲーム

<サンダーブレード>

■CZ-239AS 標準価格9,500円(税別)

© SEGA 1987

アクションゲーム

<ダウンタウン熱血物語>

■CZ-254AS 標準価格8,800円(税別)

© TECHNOS JAPAN CORP. 1989

アクションゲーム

<サイバリアン>

■CZ-229AS 標準価格8,800円(税別)

© TAITO CORP. 1988

スポーツゲーム

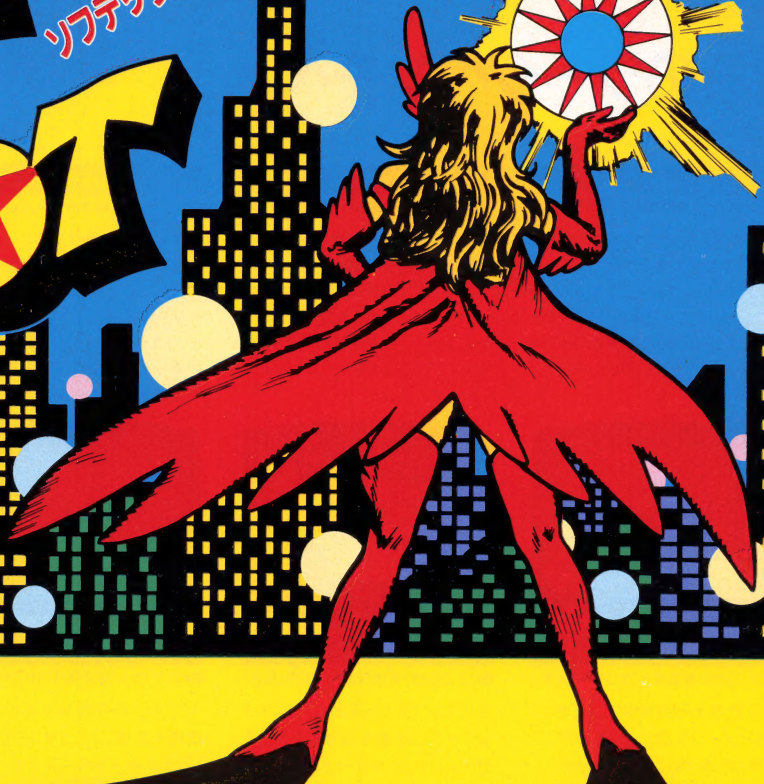
<熱血高校ドッジボール部サッカー編>

■CZ-262AS 標準価格8,800円(税別)

© TECHNOS JAPAN CORP. 1990

LUCKY ルーシーショット SHOT

ソフトック発、ゲームソフト第2弾、エキサイティング・ピンボールゲーム!



大人のゲーム「ピンボール」には、アメコミがよく似合う。

リアルな仕掛けがその気にさせる、本格的ピンボールゲーム第2弾がいよいよ登場! 美女ルーシーをはじめ、キャラ、グラフィックなど、画面いっぱい、まんまアメリカン・コミックの世界だ! マニアックな設定やマシンの動きは、キミに、ゲーセンを超えた興奮を味あわせてくれる。アウトローを気取って、ジーンズに皮ジャンで挑戦してくれ! **11月16日新発売 定価7,800円(税別)**

対応機種: PC-9801VM以降(5"2HD) PC-9801UV以降(3.5"2HD)
X68000 ※要アナログディスプレイ・FM音源対応、要16色ボード

PINBALL PINBALL

「ピンボール・ピンボール」好評発売中!

定価7,800円(税別)

対応機種:

PC-9801VM以降(5"2HD)

PC-9801UV以降(3.5"2HD)

X68000



※要アナログディスプレイ・FM音源対応、要16色ボード

eXOn

ライフ・シミュレーション

「エグゾン」

来春3月発売予定!



●物理的にシミュレートした、なめらかなボールの動き ●迫力満点アメコミ調グラフィック ●フィールドチェンジ、ブランジャヤー、4フリットパー、キックアウトホール、スロットマシン、ロールオーバーレーン、バンパー、チルトetcの趣向を凝らした仕掛けの数々。

全国通信販売

通信販売ご希望の方は、商品名、機種名、メディア名、住所、氏名、電話番号を明記の上、定価に消費税(3%)をプラスして、現金書留で下記までお申し込みください。(送料無料)

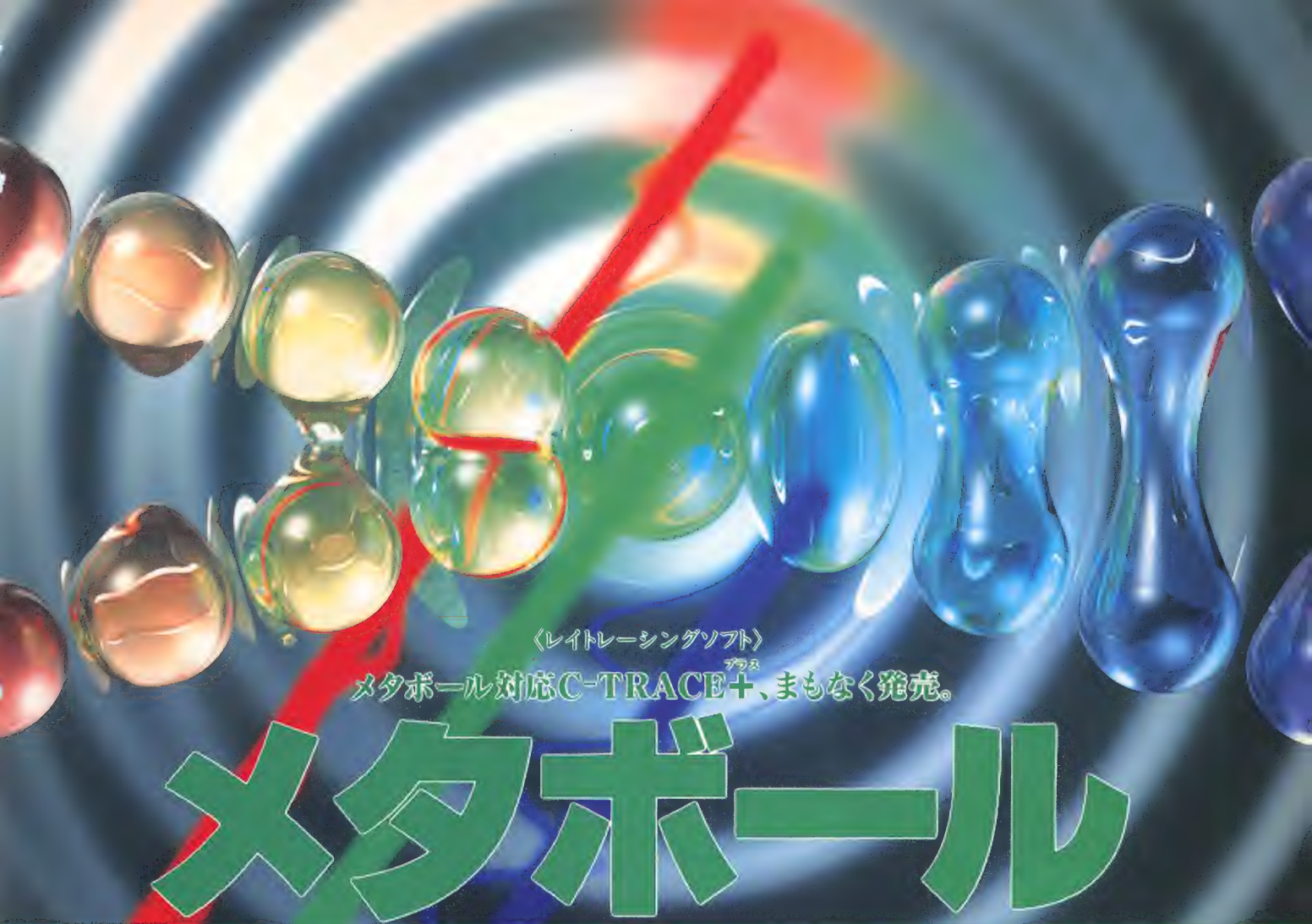
SOFTTEC

※画面写真はX68000のものです。

スタッフ
アルバイト募集中

プログラマー(PC98、X68等)、グラフィック、ミュージック、企画のスタッフを募集中です。ご連絡ください。

〒191 東京都日野市日野1161-1 カトレアマンション102
日本ソフトック株式会社 (ユーザーサポート係)
TEL.0425-82-1502 FAX.0425-87-3991



〈レイトレーシングソフト〉

メタボール対応C-TRACE^{プラス}、まもなく発売。

メタボール

メタボール…

有機的な質感を表現する事ができます。他のプリミティブと論理演算ができます。

αチャンネル機能…

αチャンネルへの対応により、高度な合成作業が可能となります。

スコープ機能…

一度生成した画像の部分修正が可能となります。

アニメーションへの応用により、作業の大幅な効率化が図れます。

スポット光源への対応…

照射範囲の設定が自由にできます。さらに照射範囲の境界のぼかしも可能です。

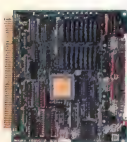
DOSEXTENDERへの対応…

CPUがV30のユーザーの方も使用できるようにノーマル版もバンドリング。(98版のみ)

対応機種…

NEC PC9801シリーズ
EPSON PC286シリーズ
EPSON PC386シリーズ
シャープ X68000シリーズ

定価…
¥198,000



レイトレーシングを高速にしたい方へ。

C-TRACE TP Ver.3.0 ¥298,000

トランスピュータボード (T-800×1+4M) + **〈ソフト+ハード〉**
C-TRACE Ver.3.0トランスピュータ版ソフトウェア
68000+C-TRACE Ver.3.0のスピードの“約170倍”
80386+C-TRACE Ver.3.0のスピードの“約40倍”

●対応機種/98版…P C 9801シリーズ、または互換機なら新旧問わずほとんどの機種に対応。ただし、標準拡張スロットがない機種には、装着できません。68版…X 68000全機種

★もっとスピードを上げたい方へ…並列処理によってスピードアップ可能！増設トランスピュータモジュール (ITRAM) ¥298,000

★フルカラーフレームバッファ	¥69,800
C-TRACE98 EXTENDER	¥128,000
C-TRACE98 Ver. 3.0	¥98,000
C-TRACE68 Ver. 3.0	¥98,000
C-TRACE TOWNS	¥68,000
C-TRACE NEWS Ver. 3.0	¥530,000
★C-TRACE98 TP Ver. 3.0	¥298,000
★C-TRACE68 TP Ver. 3.0	¥298,000
表示価格に消費税は含みません。★の製品は店頭販売いたしておりません。直接当社まで、お申し込みください。 クレジット可	



バージョンアップ受付中。

Cast

●お問い合わせ先 株式会社キャスト
〒158 東京都世田谷区等々力2-1-13
TEL.03-705-1065 FAX.03-705-5224

栄冠は君に



●5~2HD(3枚組)

12月7日発売。

標準価格 9,500円

高校野球全国大会



忘れられない熱いドラマがある。

陽が落ちるまで、夢中で千本ノックを続けたあの日の汗。初めてライバルを制した、あの練習試合の涙。

そして、厳しい予選をぐりぬけ、きょう、夢にまで見た甲子園の土を踏んだ。

さあ、球児たちの燃える視線が、あなたの采配を待っている。

彼らに栄光の優勝旗を抱かせることができるだろうか?!

感動の高校野球シミュレーションゲーム、「栄冠は君に」。

練習で選手を育て、試合で作戦を練り、約4000校のトーナメントを勝ち抜く……

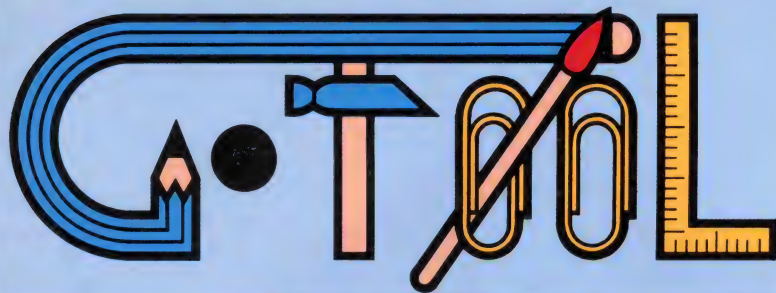
監督としてのあなたの手腕が、忘れられない夏のドラマを誕生させる！



全国3,990校の頂点を極めるのは、君だ。

* 3990校とは第71回全国高校野球選手権大会の参加総数です。
本ゲームも3990校登場します。

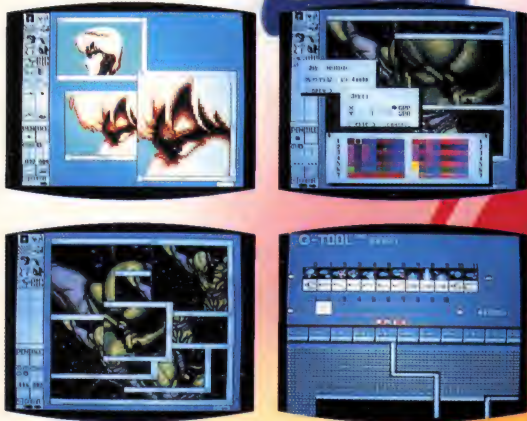
NEW CONCEPT CREATIVE TOOL



G・ツール

FOR  68000

彩先端のアートキャンバスに彩り鮮やかに感性を咲かせてください。ザインのG＝ツールは、単なるペイントツールにとどまらず、ゲームデザインをはじめとしたひとつの作品を創造する上で必要不可欠なグラフィック・キャラクタ・背景作成のすべてを備えた新感覚のグラフィックツール。驚くほどの自由さと、繊細なクリエイターのこだわりにまでアプローチしたそのコンセプトは、あなたの感性を刺激せずにおきません。



■ 概要

G・ツール	
GR-EDITモード	マルチウインドウシステム 最高12枚まで描画ウインドウが開けます。
	ユーザーアイコンシステム 使い勝手に合わせて、自分流のアイコンボードが作成可能。
	マウス機能定義システム マウスの左右に機能定義が可能。
	高速メニューウインドウ処理 メニューウインドウの開閉が瞬時に行えます。
BG-EDITモード	スプライト処理 作成から修正、アトリビュートが行えます。
	スプライトカラー処理 16色に分まとめて処理できます。
	背景の作成 最大250画面分を自由に設定することができます。
	キャラクタチェック機能 単独チェックのほか、背景と重ねてのチェックも可能。

ア
ー
ト
セ
よ。

好評
発売中



■ 5"2HD
定価¥28,000(税別)

zainsoft

株式会社 ザインソフト 〒651 神戸市中央区磯辺通2丁目2-10新南泰ビル10F TEL.(078)242-2855(代表)

資料・オートデータ
請求書
0714
12.21号

△68000

近日発売!!

この扉の向こうに...



おどろおどろしいモンスター
奥深い謎解き
興味天外な民
幽霊屋敷から始まる
不気味なRPG
すべてに恐ろしい
あの安田 均氏のホラーワールドは
ゲームを終えた後が本当に恐ろしい。



本格ホラーRPG ゴーストハンターシリーズ#1
ラプラスの魔
原作 安田 均 音楽 小坂 明子



●角川文庫 ●コンパコミックス ●ビジネスアスキー

©Kadokawa shoten/H.YASUDA & Group SNE
MSX マークはアスキーの商標です。

ユーザーステレホン ☎大阪06(315)8255

平日の午後1時半から6時の間は、お問い合わせに直接お答えします
その他の時間と土・日・祝日はまるまる24時間録音できるテープサービスです

◆標準価格に消費税は含まれておりません。お買上げの際に別途消費税をお支払い下さい。
◆通信販売ご希望の方は、住所・氏名・電話番号・商品名・機種名・メディアを明記の上、現金書留または郵便振替(大阪8-303340)にてお申し込み下さい。送料は無料ですが、標準価格に消費税の3%を加えた金額をお送り下さい。



Humming Bird Soft™

株式会社エム・エー・シー ハミングバードソフト
〒530 大阪市北区曽根崎2丁目2番15号

怪魔退散!!

踏み出すとそこは殺戮の地。
勇剛者達よ、立ち上がれ!!

U N D E A D L I N E

幻獣鬼

けんじゅうき

△68000

◀5"2HD・3枚組 標準価格¥7,800 (税別)▶

ACT-neXt

はじめからひとつひとつステージをクリアしていくのが、ほとんどのアクションゲームの形態。クリアするたびに難度があがってなかなか先のステージにチャレンジできない。そこであきらめてしまったり、「だから、プレイしたくないんだ、苦手なんだ」という人もいるだろう。この課題に取り組み、解決したのが、ステージ・セクション・システム、略してS・S・S。8ステージのどのステージからでもプレイ可能。好きなように選択して楽しめる。だからといって、初心者のゲームと勘違いしては困る。「アクションゲームは正攻法にプレイすべきだ」と思っている人には、難度を非常に高く設定した「HARD」を用意している。どのくらいのゲーマーが最後まで到達できるか、楽しみだ。

- ★プレイヤーは、戦士、魔道士、忍者の3キャラクターから選択
ゲーム途中で一度クラスチェンジが可能
- ★各キャラクターは、通常武器の他にオプション武器が3種類装着可能
- ★ジョイスティック対応
- ★FM音源とADPCMに対応
- ★ゲーム中の全曲が聴けるミュージックモードあり

neXt

RPG・ACT・SLG、最強のラインナップで
次世代傳統



RPG-neXt

ルーンワース 黒衣の貴公子

- X68000
- PC-9801VM、UVシリーズ
- PC-286、386シリーズ、NOTE対応
- PC-8801SRシリーズ・VA、98DO対応
- MSX2/MSX2+

標準価格 各¥8,800 (税別)



NEW 3D GOLF SIMULATION

遙かなるオーガスタ

オーガスタ・ナショナル・ゴルフ・クラブと正式契約

SLG-neXt
遙かなるオーガスタ

発売
待機中



■通信販売をご希望の方は、現金書留で料金と商品名・機種名と電話番号を明記の上、当社宛にお送りください。(速達希望の方は300円プラス)

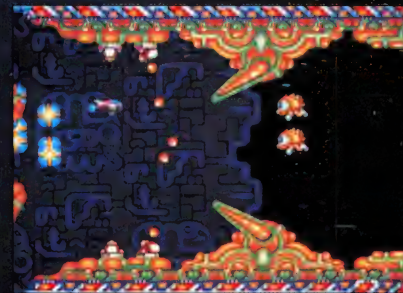
●T&Eの最新情報がわかるテレフォンサービス
Phone052-776-8500

Technology & Entertainment Software

T&ESOFT

株式会社 ティーアンドイーソフト

〒465 名古屋市中東区豊が丘1810 PHONE: 052-773-7770

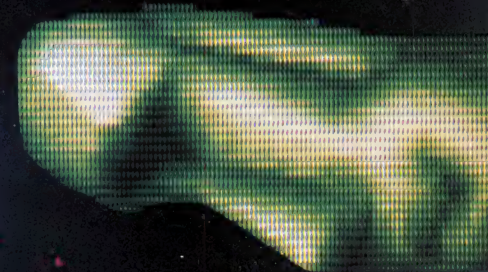


※この画面は開発中のものです。

PARODIUS パロディウスだ!™

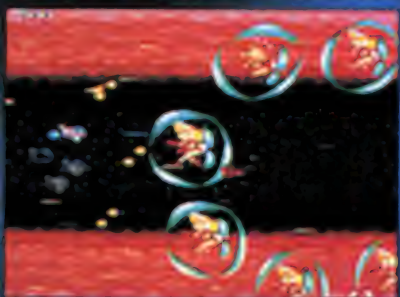
— 神話からお笑いへ —

© KONAMI 1990



笑って砕けよ。

あああ〜っ、かくも恐ろしき世界よ。
二度と、私の知性を乱してくれるなどあれほど頼んだのに、
再びその滑稽な姿かたちを現したな。
しかも、ゲームセンターで我が物顔をしていた
そのまんまの容姿ではないか。
ここまでやってしまったのか、というパロディウス倫理委員会の
ささやきを縦横無尽に交わしながら、
ほぼ完全なる移植にむけ着々と進攻している。
X68だから、よけいにこわい。
もうプレイヤーは体当たりで砕けちるしかないのだろうか。



 68000シリーズ

コナミ株式会社

本社／〒101 東京都千代田区神田神保町3丁目25 大阪支店／〒541 大阪府豊中市庄内栄町4丁目23番18号 札幌支店／〒060 札幌市中央区北1条西5丁目2番5号 福岡支店／〒810 福岡市中央区天神2丁目8番30号

リアルティ。



※この画面は開発中のものです。

感動の

「野球ゲームに革命を！」これが始まりだった。リアリティをどこまで追求できるか。だれも考えつかなかった映像とは？ここに、技術と遊び心の髄を極めた新しい野球時代が訪れる。その名は生中継68。アングルはテレビ実況中継そのものの体乗り出し、手に汗にぎるあの興奮がお茶の間を大スタジアムへと変えてしまう。野球にシーズンという言葉は消えるだろう。



生中継68

© KONAMI 1991 TM

 680000に登場

宇宙は、野望だけでは支配できない。

宇宙暦796年、銀河系はゴールデンバウム王朝が支配する銀河帝国と、その専制政治に反対する自由惑星同盟の両陣営が激しい戦闘を繰り返していた……。圧倒的支持を得た「銀河英雄伝説」を遥かに凌ぐスケールで、今新たな伝説が生まれようとしている。銀河英雄

伝説IIだ。帝国軍の若き天才ラインハルト、そしてヤン・ウェンリーの熱い闘いが、再び始まる。星系マップは従来の4倍、3Dグラフィックによる戦闘シーンなど、あらゆる面でパワーアップされている。田中芳樹原作の大人気スペースオペラ「銀河英雄伝説」。宇宙の歴史を変える闘いは、ここに始ろうとしている。

SPACE WAR SIMULATION

銀河英雄伝説II

銀河英雄伝説II X68000シリーズ 11月30日発売 ¥9,800 (税別)

- 5.2HD(4枚組)●X68000専用グラフィック●2重スクロール●MIDI音源対応
- FM音源・ADPCM対応●HMS(HYPER MOUSE SYSTEM)搭載

イラスト 加藤直之 ©1990 BOTHTEC ©1990 QUEST ©1990 Micro Vision ©1988 田中芳樹・徳間書店・徳間ジャパン・キティフィルム

BOTHTEC®

ボーステック・ソフトウェア

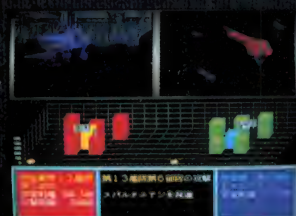
株式会社クエスト(旧ボーステック株式会社) 〒158 東京都世田谷区用賀2-18-8 TEL.03-708-4711

スタッフ募集 ゲーム企画・プログラマー・音楽担当者

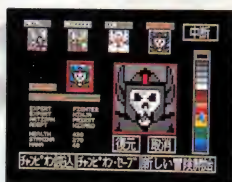
採用係連絡先 TEL.03-708-4712



※通信販売(送料サービス)：品名・機種・住所・氏名・電話番号を明記して、現金書留でお申し込みください。



画面は開発中のものです



勇者たちよ!

いま一度

新たな挑戦の幕開けが迫る!

■世界をそして日本を興奮の渦に巻き込んだリアルタイムRPG「ダンジョンマスター」。その続編が帰ってきた!「ロード・カオス」を倒し世界に平和と秩序をもたらした勇者達…。しかし「ロード・カオス」は生きていた。あの時、彼は既に自分が勇者達に倒されることを知り、秘かに新しいダンジョンを作った。悪の力を蓄えていたのだ。

■アニメを使ったイントロダクション、FM音源対応による音楽とサウンド、チャンピオンの顔を自由に書き換えることのできるキャラクターエディット機能、さらには日本版独自のプレイヤーの現在地を一目で表示するマップ表示機能や新しい魔法の追加など熱中度さらにパワーアップ。

■前作で育てた勇者をそのまま使って冒険に旅立つもよし、新たに用意された勇者を編成してダンジョンに向かうもよし、ドキドキしながらこの興奮を味わって下さい。前作以上の難しさとおもしろさだけは保証します。

ダンジョンマスター

CHAOS STRIKES BACK

戦いは終わらない。

いよいよ
発売!

続

ダンジョンマスター | カオスの逆襲

■ 12月中旬発売 X68000版 ¥9,800(税抜)

〈注: 24KHzモード対応モニターならフル画面表示〉

© 1990, SOFTWARE HEAVEN, INC./FTL GAMES. LICENSED THROUGH AN AFFILIATION WITH J.P. INTERNATIONAL. © 1990 VICTOR MUSICAL INDUSTRIES, INC.

発売: ビクター音楽産業株式会社

企画・開発: Fill in Cafe



シューティングの極み!

ゲーム性、グラフィックス、サウンド何もかもがX-68000の限界を超えた!!

- 驚異の迫力で展開する変化に富んだ全10ステージ。
- 各ステージ毎に武器(5種類)の選択、ステージの解説の表示をし、その間のプログラムのロードにより途中のディスクアクセスがなく、スムーズにプレイできます。
- 大型エネルギー・ゲージ、スコア・ゲージの採用による迫力ある戦闘がたのしめます。
- 戦闘中のスピード変更機能搭載による状況に応じた臨機応変の戦いを実現。
- 快感のテーマ曲ほか全20曲収録。内蔵音源に加えてMIDI音源にも対応。迫力のサンプリング効果音も搭載。



11月23日
発売

本格的3D快感
シューティングゲーム

ニューラル・ギア

●X-68000対応 ¥8,800(税抜)

宇宙が理性を挑発する

STORY

星暦3960年、シュヴァルツシルト銀河外縁部シロ星団には大小さまざまな国々が林立していた。そして、物語はシロ星団の南西部に位置する“サンクリ星団”から始まる。時にKGD星域に遊学中であったサンクリ星国皇太子は、惑星ウーリィに行幸中の父王の暗殺、そして惑星ウーリィの反乱という相次ぐ凶報に、急ぎ帰国の途に付いた。そして、慌ただしく即位式を済ませた後、反乱鎮圧と父王の仇を報じる事に新王の威信を賭けることとなるのである。

シュヴァルツシルト・X68000版
12月上旬発売予定。

■通信販売(送料無料)のお知らせ

工画堂スタジオでは通信販売をしております。

ご希望の方は、品名・機種名・住所・氏名・電話番号を明記の上、3%の消費税を加算して現金書留でお申し込みください。

ストーリー性を持ったドラマティックなゲーム展開

シュヴァルツシルトの最大の特徴は、そのゲームシステムにあります。単なるウォーシミュレーションではなく、ゲームを進めていくにしたがって、次々に新たな目的が現われ、プレイヤーは知らず知らずにゲームのシナリオに引き込まれていくという、ドラマティックなゲーム展開が魅力の、SFシミュレーションゲームです。

究極のゲームシナリオ

ゲームのおもしろさはシナリオで決まります。軍事行動、外交政策調査・研究、資金運用、商業取引引きといった戦略要素を完璧にシミュレート。シミュレーションゲームの面白さを徹底的に追求した究極のゲームシナリオです。



SCENARIO SIMULATION GAME

狂嵐の銀河

Schwarzschild

シュヴァルツシルト

●5"2HD・2枚組 ¥12,800 (価格には消費税は含まれておりません)

■開発スタッフ募集のお知らせ
プログラム・アシスタントプログラム・ゲームデザイン・グラフィックのスタッフを募集中です。
御連絡ください。

KOGADO
Software Products

〒162 東京都新宿区市谷台町11
TEL. 03-353-7724

資料請求券

On!X-12月号

X68000

MIDI対応

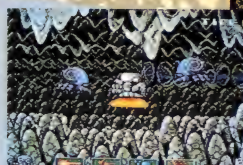
全方向スクロール、
ロボットシューティングの極限!

アーケード版アトミックロボキッド、
X68000に登場!
'90年年末、ついにその全貌を現わす。

ATOMIC ROBO-KID

アトミック・ロボキッド

12月
発売予定



2XXX年、核戦争の難を逃れた少数の人々は、地下に隠れて放射能の嵐がやむのを待ち、地上に出た。しかし、人々のDNA(染色体)は、わずかに残った放射能によって破壊され、人類は自分達の子孫を残せなくなってしまう。トミタ博士は、わずかに残された人々をシェルターに冷凍保存しDNA正常化のプログラムを開発する。ところが、シェルターに向かうロボキッドが動き出す直前に博士は、死んだ。自分の目的も解らずに、目覚めたロボキッド。……はたして、ロボキッドは人類を救う事ができるのか?

X68000対応 5"-2HD

●ローランド社

MT-32、CM32L、CM64完全対応

MIDIインターフェイスボードC-Z-6BMI
又は、SACOM製SX-68Mが必要です。

標準価格8,800円

copyright
©UPL

密林深く眠る失せし文明に、
蘇る血の運命。すべての謎は、

魅由の繰り広げるミステリアスアニメーションアドベンチャー第2弾!!

闇の血族

THE PREDESTINED HOMICIDES #2

一人の少女の下に今、
遥かなる真実を紡ぎだす。

古えの封印は解かれ、時間の糸車は無数の運命の糸を引き、血染めの歴史絵を紡ぎはじめた。一連の殺人事件に秘められた暗号を解く唯一の手掛かりを求めて、魅由は親友の理沙と共に中米の地へとおもむく。そこには思いもよらぬ宿命の翼が待ち受けていたのであった。

1990年6月



完結編

アーケードフィールド宣言!!

ジェミニウイング Gemini Wing

MIDI対応

アドレナリン全開シューティング
「ジェミニウイング」

Now on sale!!

スクランブル 「ガンシップ緊急発進!!!」

君は迫り来る「蟲」の恐怖に耐えられるか?
そして最後に持ち受ける真の敵の正体とは!?

ジェミニウイング オリジナルブルゾンプレゼント!!

通信販売でジェミニウイングをお買い求めの方、
先着100名様に限りオリジナルブルゾンをもらえるプレゼント!!

お申込は、現金書留に商品名、住所、氏名、
電話番号を明記の上¥9,064(税込み)
を当社までお送り下さい。

X68000対応 5"-2HD

●ローランド社

MT-32、CM32L、CM64完全対応

MIDIインターフェイスボードC-Z-6BMI

又は、SACOM製SX-68Mが必要です。

標準価格 8,800円

copyright
©TECMO



NOVEL WARE

MIDI対応



FM-TOWNS版 12月発売予定
¥12,800[上下巻1セット]

X68000対応 5"-2HD

●ローランド社

MT-32、CM32L、CM64完全対応

MIDIインターフェイスボードC-Z-6BMI

又は、SACOM製SX-68Mが必要です。

標準価格 8,800円

※標準価格には消費税は含まれておりません。

株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
ハードウェア部 03(635)5145
ソフトウェア部 03(635)7609



ACTION GAME

DYNAMITE



ダイナマイト・デューク



奴の野望は、俺が打ち砕く!!

西暦2089年、オゾン層の破壊による環境悪化により、人類は滅亡の危機に瀕していた。

この事態を憂慮した各国の指導者は、世界連合軍を設立、厳しい環境下にも耐える人体の研究開発に乗り出す。

そして、ついにバイオテクノロジーの力によって、強化人間を実現させたのである。

しかし、開発に関わった十人の大佐が技術を盗み出し、強化人間をも超える存在を創り上げることに成功。

……、彼の目的は、軍隊を組織し人類を支配することであった。

連合軍は、この野望を打ち砕くため、ひとりの強化人間を送り出した。


機械の右腕を持つ男、陸軍特殊部隊大佐デューク・フリードリッヒ・フェルゼン、コード・ネーム=Red・Dynamite。

しかし、人は彼のことを、こう呼んだ「ダイナマイト・デューク!!」。



●大人気アーケード・ゲームからの移植／●シューティングの興奮+アクションの一体感／●アメリカン・コミック調の美しい画面／●アニメのように派手に動く迫力デカ・キャラ／●変化に富んだ各ミッションと個性あふれる敵キャラ／●必殺の一撃「ダイナマイト・パンチ」の快感!

90年11月30日発売!! X68000専用(ジョイスティック・キーボード対応FM音源対応1PLAYER専用) 5"2HD 3枚組 ¥8,800(税別)

 株式会社 ヘルツ

〒169 東京都新宿区北新宿2-1-16松本ビル3号2F TEL.03(371)3012

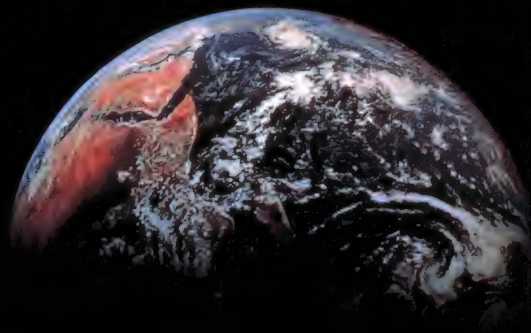
©1989 SEIBU-KAIHATSU INC.

STAFF 募集 !!

企画、プログラマー、デザイナー、サウンドクリエイター
詳細は電話連絡にて。担当秋山まで。



救え!!



地球を



緊急事態発生!!

20××年、東西両陣営の対立を背景に、西側ムーンベースの大爆発が起こった。原因究明を急ぐ中、偵察衛星は驚くべき事実を写しだしていた。ムーンベースのマザーコンピュータに寄生する妖しげな植物……。果してエイリアンの侵略なのか。その時、西側の最新鋭戦闘機OF-1『ダイダロス』にスクランブル指令が下った。過酷なシミュレーションファイトをクリアした精鋭達が、東西の壁を越え、地球を、人類を救う戦いへと、今、飛び立とうとしている。

Image Fight™

特別付録

X68000版
オリジナルテレホンカード

■5"2HD2枚組

■ジョイスティック対応

■難易度設定(4段階)可能

■フルオート連射機能内蔵

X68000イメージファイト

12月14日発売予定

予価9,700円(税別)



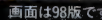
※ アイレムファンクラブ会員募集中。詳しくは06-534-1060もしくはお近くのテレホンサービスまで。

テレホンサービス 札幌 011-685-9131 名古屋 052-323-0270
福岡 092-475-9400 東京 03-823-5130 大阪 06-535-0651

アイレム株式会社

〒550 大阪市西区阿波座2丁目2番18号 西本町川洲住生ビル13F

A riddle of parallel world Roll-Playing game.



- レイ・ガンは、エルフ初の本格的フィールドタイプのRPGです。スタッフが燃えに燃えて制作しました。
- レイ・ガンでは、これまでエルフ初　16色アナログ　表示です。女子の肌はあくまでも美しくきれいに、モンスターは中間色をまんだんに使用し、よりリアルに描かれています。ご期待ください!!
- 戦闘シーンも本格的です!! 画面だけではわかっていただけないのが残念です。効果音、ビジュアルに気を使っているほど使い、見ていただいても楽しめるような戦闘シーンを作りました。
- モンスターの種類ももりだくさんです。次にどんなモンスターが出てくるか……ワクワク、ドキドキしながらプレイしてください。
- エルフ自慢のグラフィックがさらにパワーアップ!! 16色のアナログはもちろんの事、キャラクターだけでなく、背景にも力を入れて描いています。もちろんメインとなるグラフィックは画面の2/3を占めるビックなサイズです!!
- アニメーションも今回は凝りました……とにかくスタッフ全員で「大きく動かそう!!」をモットーに作り上げました。
- 今回も好評のオマケディスクがついています。(オマケディスクとは? =自分が今までクリアした所までのビジュアル(ちなみに女子ののお楽しみ画面がメインです。)を何度でも手軽に見れるモード)です。
- 98、X68000版の3枚組、80版では8枚組(予定)、MSX版では6枚組(予定)というゲームサイズで、お値段かな、なんと6,800円!!……そう、エルフでは一人でも多くのお客様に、(レイ・ガン)を楽しんでいただきたいのです。

●現金書留の場合……
商品名、機種、メディアを明記の上
エルフまでお送り下さい。
●郵便振替の場合……
郵便局の振替用紙に商品名、機種、
メディアを明記の上、口座番号
東京3-191196
エルフまでお申し込み下さい。



全 国 通 販

SHARP 認定
PPO-SHOP

O.A.ランド

(TEL) **03-770-8855**

- アフターサービス万全のサポート体制
- 下取・買取は電話で見積りしております。責任を持って下取りさせていただきます。
- ご注文、お問合せは…午前10時から午後7時まで
- 商品のお届けは…入金確認後、即日発送致します。
- TEL・FAXのお見積OK!!
- 低金利クレジットをご利用下さい。

▶11・15～12・14

SHARPのことなら

なんでおまかせ!!

ボーナス・シーズン大徳買セール! 安く値切ってネ。

お電話下さい。価格をお知らせいたします。

SHARP X68000シリーズセット

●次代のインテリジェンス= SX-WINDOW搭載!!

X68000 EXPERT II

- CZ-603C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計

¥453,000

クレジット 12回 ¥30,100 24回 ¥15,900 36回 ¥11,000



X68000 EXPERT II-HD

- CZ-613C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計

¥563,000

クレジット 12回 ¥37,400 24回 ¥19,800 36回 ¥13,700

X68000 PRO II

●SX-WINDOW搭載!!

- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計

¥400,000

クレジット 12回 ¥26,600 24回 ¥14,000 36回 ¥9,700



X68000 PRO II-HD

- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計

¥510,000

クレジット 12回 ¥33,900 24回 ¥17,900 36回 ¥12,400

X68000 SUPER-HD

●SX-WINDOW搭載!!

- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック

クレジット

12回 ¥42,100 24回 ¥22,300 36回 ¥15,400



X68000 SUPER-HD

- CZ-623C-TN(チタン)
- CZ-613D-TN(チタン)
- MD-2HD 20枚

定価合計

¥633,000

セットで購入のお客様に、ディスク(10枚)、ゲームバックサービス中!!
さらに、期間中ゲームソフトが1本付きます。詳しくは、お電話下さい。

SHARP フリートップパーソナルコンピュータ

電子手帳だよ〜ん便利です!!

AX286N-H2(MZ-8376A)

- ①Business Mate 標準装備
- ②20MB+HDD
- ③フリートップサイズ
- ④世界標準AX仕様
- ⑤内部専用スロット



- ⑥優れた拡張性
- ⑦SPシステム 標準装備

定価 ¥398,000
大特価!!
電話で値切ろう!!

①PA-9500 NEW

………▶大特価!! TEL下さい

②PA-8500 台数限定

………▶大特価 ¥15,000

③PA-7500 台数限定

………▶大特価 ¥12,000

●数に限りがございますので、早目にTEL下さい。

新製品 周辺機器

光磁気ディスクユニット

- CZ-6MO1 (定価 ¥450,000)

特価 ¥360,000

SCSIボード

- CZ-6BS1 (定価 ¥29,800)

特価 ¥24,800

X BAS to C CHECKER PRO68K

- CZ-260LS (定価 ¥9,800)

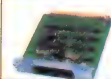
特価 ¥8,000

流通事情により、広告表示価格は、
お安くなる場合がありますので、ドンドンお電話下さい。



CYBER STICK
■CZ-8NJ2
(定価 ¥23,800)

OAランド特価
▶**¥18,000**



MIDIインターフェースボード
サコム (定価 ¥19,800)
■SX-68M
●X68000専用純正コンパチ

OAランド特価
▶**¥15,800**

周辺機器コーナー

プリンターセットコーナー

①CZ-8PC4 (GY) (48ドット/カラー対応/ハガキ可能)
定価 ¥99,800 …… **特価 ¥64,800**

②CZ-8PK10 (24ピン漢字プリンター136桁)
定価 ¥97,800 …… **特価 ¥78,000**

③CZ-8PG1 (24ピンカラー漢字プリンター80桁)
定価 ¥130,000 …… **特価 ¥103,000**

④CZ-8PG2 (24ピンカラー漢字プリンター136桁)
定価 ¥160,000 …… **特価 ¥125,000**

OAランド特選品!!



■IO-735X (定価 ¥248,000)

●カラーイメージ
ジェットプリンター

特価 ¥190,000

モデム

X68000周辺機器コーナー

オムロン MD-1200A III	¥14,500
MD-24FP4 II	¥27,500
MD-24FP5 II	¥29,800
MD-24FN4	¥28,000
MD-24FN5	¥31,300
MD-24FJ4	¥31,300
MD-24FJ5	¥34,500
MD-24FS4	¥28,500
MD-24FS5	¥34,500
アイワ PV-A24VM5	¥32,500
PV-M24	¥28,800
NEC COMSTAR 2424/4	¥28,800
COMSTAR 2424/5	¥33,500

①CZ-6VT1 (カラーイメージユニット)	定価 ¥69,800 …… 特価 ¥52,500
②CZ-6NS1 (カラーイメージスキャナ)	定価 ¥88,000 …… 特価 ¥141,000
③CZ-6BM1 (MIDIボード)	定価 ¥26,800 …… 特価 ¥20,500
④CZ-8NJ2 (インテリジェント・コントローラー)	定価 ¥23,800 …… 特価 ¥18,000
⑤CZ-6TU (RGBシステムチューナー)	定価 ¥33,100 …… 特価 ¥25,000
⑥CZ-64H (増設ハードディスク)	定価 ¥120,000 …… 特価 ¥90,000
⑦CZ-6EB1 (拡張I/Oボックス=4スロット)	定価 ¥88,000 …… 特価 ¥66,000
⑧CZ-6BP1 (数値演算プロセッサボード)	定価 ¥79,800 …… 特価 ¥60,000

■I-O DATA 増設RAMボード

●1MB増設RAMボード
PIO-6BE1-A

定価 ¥25,000



●2MB増設RAMボード
PIO-6BE2-2M

定価 ¥50,000



●4MB増設RAMボード
PIO-6BE4-4M

定価 ¥88,000



特価 ¥19,000 特価 ¥36,300 特価 ¥64,000

OAランド今月の大玉!! = 超A級中古品

◎1年間完全保障、新品同様(美品) = お問い合わせ下さい。

■SUPER-HDセット	3セット限り
CZ-623C-TN + CZ-613D-TN (定価 ¥633,000) ……	特価 ¥445,000
■EXPERT-IIセット	3セット限り
CZ-603C-BK + CZ-605D-BK (定価 ¥453,000) ……	特価 ¥310,000
■PRO IIセット	2セット限り
CZ-653C-BK + CZ-605D-BK (定価 ¥400,000) ……	特価 ¥275,000

OAランド推奨 周辺機器

■SX-WINDOW

(次代インテリジェント・ソフト)
●多機能コントロールパネル搭載の
本格ウィンドウシステム。

定価 ¥6,800

特価 ¥5,100

■CZ-6BV-1

(ビデオボード)
●ビデオ出力は、テレビ標準
信号、拡張I/Oスロット使用

定価 ¥21,000

特価 ¥15,800

■CZ-245LS

(C-コンパイル)
●ソースコードデバッグをはじめ、
各種開発ツールを強化。II版

定価 ¥44,800

特価 ¥34,000

通信販売のご案内

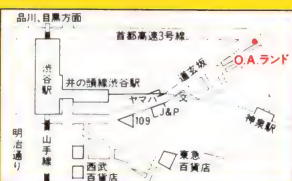
全国通販

■銀行振込で申し込みの方は商品名
及びお客様の住所・氏名・電話番号
をお知らせ下さい。

[振込先]第一勧業銀行 渋谷支店

普通No.1163457 株式会社オーランド

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。
■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円より自由に設定できます。



■12月より年内無休!!

クレジット表

3回	3%	6回	4%	10回	5.5%	12回	5.5%	15回	8%	18回	10%	20回	11%
24回	11.5%	30回	15.5%	36回	16%	42回	20.5%	48回	21%	54回	26.5%	60回	27%

株オー・エー・ランド

〒150 東京都渋谷区円山町20-4 第5日新ビル1F

☎(03)770-8855

FAX (03)770-7080

関東エリアの送料は、1個につき¥1,000です。

★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合わせ下さい。掲載の価格は、10月下旬現在です。

ちゃんとパソコンしたい人の。

TSUKUMO



もう、
離れられなくなるね。

あなたに、会いたい。

近日 **OPEN** 予定

うわさのパソコンロフト「ツクモパソコン本店」オープン

日本で初めての総合カメラ専門店「ツクモAV/カメラ館」はオープンセール中

PRESENT ツクモ全店で1万円以上お買い上げの方先着1万名様に越智静香チャンのフロッピーカレンダーをさしあげます。
静香チャンに会おう! 12月24日(月)PM1:00～AV/カメラ館5Fのイベントフロアにて越智静香チャンのサイン会が開かれます。

ツクモパソコン本店 **ツクモAV/カメラ館**

〒101 東京都千代田区外神田1-9-7 ☎03-253-5599

〒101 東京都千代田区外神田1-11-3 ☎03-254-3999

パソコン本店OPEN/パソコン本店2FにX68000コーナーが移動してさらに充実! X68000ファンに伝えます。

掲載商品
2万円以上
送料無料!!

A Happy New Open!

コンピュータでクリスマスカード&年賀状を♡♡♡

(SOFT) ★CANVAS PRO-68K
(CZ-249GS) 定価 ¥29,800

●CANVAS PRO-68K ドローグラフィックライ
ブラリVol.1・2 定価各 ¥8,800

★NEW PrintShop PRO-68K
(CZ-221HS) 定価 ¥19,800

(HARD) 台/数/限/定

★24ピンドットプリンタ (80桁)
Happy特価 ¥39,800
(消費税別 ¥1,194)

★48ピン熱転写カラー漢字プリンタ
Happy特価 ¥59,800
(消費税別 ¥1,794)

ハードディスク目玉品

40MB SASIタイプ
Happy特価 ¥59,800
(消費税別 ¥1,794)

光磁気ディスクユニット 台数限定

ソニー NWP-539N (光磁気ディスクユニット) ¥440,000
シャープ CZ-6BSI (SCSIボード) ¥29,800
SCSIケーブル ¥5,000
光磁気ディスク サービス (¥30,000)

合計定価 ¥509,800

Happy特価 ¥408,000 (消費税別 ¥12,240)

クレジット例 (48回払・税込) 初回 ¥13,110 + 月々 ¥11,300 × 47回

△68000 シリーズ



PROI CZ-653C 定価 ¥285,000

CZ-663C 定価 ¥395,000

EXPERT I CZ-603C 定価 ¥338,000

CZ-613C 定価 ¥448,000

SUPER HD CZ-623C 定価 ¥499,000

本店オープン記念特価にて
提供中!
是非おたずね下さい。

ミュージックツールNEWプラス

Aセット

CM-32L ¥69,000
SX-68M ¥19,800
Musicstudio Mu-1 ¥19,800
合計定価 ¥108,600
Happy特価 ¥88,000
(消費税別 ¥2,640)

クレジット例 (18回払・税込)

初回 ¥7,223 + 月々 ¥5,600 × 17回

※「Musicstudio PRO-68K V1.1」又は「MUSIC PRO-68K (MIDI)」のソフトに変更の場合は ¥8,000 プラスになります。但し、これらのソフトがバージョンアップされた場合には変更する場合がございます。

追加オプション機器

●はなうたくん CP-40 定価 ¥33,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

●MIDIキーボード・コントローラー PC-200 定価 ¥36,000

お得なM-ローセット

アートツール

ハードウェア 限定品

一流メーカーイメージスキャナ
Z's STAFF PRO-68K Ver2
Happy特価 ¥49,300
(消費税別 ¥1,479)

マジックパレット
Happy特価 ¥16,800
(消費税別 ¥504)

サイクロンExpress α88
Happy特価 ¥83,300
(消費税別 ¥2,499)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

デジタルクラフト
Happy特価 ¥38,800
(消費税別 ¥1,164)

コミュニケーションツール

モデム 一流メーカー 2400ボーラス4 超・特・価
Happy特価 ¥28,000 (消費税別 ¥840)

通信ソフトウェア
た〜みのる2 Happy特価 ¥15,000
Communication PRO-68K Ver.2.0 定価 ¥19,800

ハードディスク

★SASIタイプハードディスク
アイテック IT X840 特価 ¥84,800 (消費税別 ¥2,544)
アイテック IT X880 特価 ¥99,800 (消費税別 ¥2,994)
(カラー・ブラック・グレー)
★SCSIタイプハードディスク
アイテック IT X80S 定価 ¥128,000 特価 ¥108,000
(消費税別 ¥3,060)
アイテック IT X130S 定価 ¥158,000 特価 ¥128,000
(消費税別 ¥3,840)
(CZ-6BSI SCSIボードは別売 定価 ¥29,800)

光磁気ディスクユニット

シャープ CZ-6MO1 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

シャープ CZ-6BSI 特価 ¥108,000 (消費税別 ¥3,060)

注目!!

翌月一括払い(12月末)はもちろん
冬のボーナス一括払い
手数料(金利)無料
(平成3年1月末払いをご利用下さい。)

プリンター 10台限定 (送料¥1,000)
■CZ-8PK8 (定価¥152,000)
●24ピン漢字プリンター P&A
(136桁) 限定特価¥49,800
●ハガキ印字OK!! (送料・消費税込 ¥52,324)

CYBER STICK

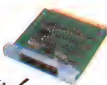
●CZ-8NJ2
(定価¥23,800)
超特価!!

¥18,500 (送料・消費税込み ¥19,570)



X68000シリーズ専用 特価¥16,480

MIDIインターフェースボード
SX-68M (サコム)
(純生コンパチ) 定価¥19,800
送料・消費税込み!!



またまた

秋葉原でおなじみの

11/15~12/14

- お近くの方はお
- 本体単品で特
- ビジネスソフト定

X68000用メモリーボード (I/O・DATA) (送料¥500)



①PIO-68E1-A ¥18,000
(送料・消費税込 ¥19,055)
②PIO-68E2-2M ¥36,500
(送料・消費税込 ¥38,110)
③PIO-68E4-4M ¥64,300
(送料・消費税込 ¥66,744)

ジョイスティック 送料 ¥500
●X-1PRO
定価 ¥9,500 ▶ 特価 ¥7,800
●ASCII STICK
定価 ¥6,800 ▶ 特価 ¥5,500

NEW X68000EXPERT II/II-HD & PROII/PROII-HD & SUPER-HD (送料・消費税込)



EXPERT II

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ジョイカード2枚 }

EXPERT II-HD

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ジョイカード2枚 }



PROII

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ジョイカード2枚 }

PROII-HD

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ジョイカード2枚 }

SUPER-HD

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ジョイカード2枚 }

EXPERT II

(A)セット: CZ-603C+CZ-604D	定価 ¥432,800 ▶ 特価 (価格はお電話下さい)
12回 26,800 24回 14,100 36回 9,800 48回 7,600 60回 6,400	
(B)セット: CZ-603C+CZ-605D	定価 ¥453,000 ▶ 特価 (価格はお電話下さい)
12回 28,300 24回 15,000 36回 10,400 48回 8,100 60回 6,800	
(C)セット: CZ-603C+CZ-613D	定価 ¥473,300 ▶ 特価 (価格はお電話下さい)
12回 29,800 24回 15,700 36回 10,900 48回 8,500 60回 7,100	
(D)セット: CZ-603C+CU-21HD	定価 ¥486,000 ▶ 特価 (価格はお電話下さい)
12回 30,500 24回 16,100 36回 11,100 48回 8,700 60回 7,300	

EXPERT II-HD

(A)セット: CZ-613C+CZ-604D	定価 ¥542,800 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(B)セット: CZ-613C+CZ-605D	定価 ¥563,000 ▶ 特価 (価格はお電話下さい)
12回 36,300 24回 19,200 36回 13,300 48回 10,400 60回 8,700	
(C)セット: CZ-613C+CZ-613D	定価 ¥583,300 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(D)セット: CZ-613C+CU-21HD	定価 ¥596,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

PROII

(A)セット: CZ-653C+CZ-604D	定価 ¥379,800 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(B)セット: CZ-653C+CZ-605D	定価 ¥400,000 ▶ 特価 (価格はお電話下さい)
12回 25,100 24回 13,300 36回 9,200 48回 7,200 60回 6,100	
(C)セット: CZ-653C+CZ-613D	定価 ¥420,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(D)セット: CZ-653C+CU-21HD	定価 ¥433,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

PROII-HD

(A)セット: CZ-663C+CZ-604D	定価 ¥489,800 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(B)セット: CZ-663C+CZ-605D	定価 ¥510,000 ▶ 特価 (価格はお電話下さい)
12回 32,900 24回 17,400 36回 12,100 48回 9,500 60回 7,900	
(C)セット: CZ-663C+CZ-613D	定価 ¥530,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(D)セット: CZ-663C+CU-21HD	定価 ¥543,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

SUPER-HD

(A)セット: CZ-623TN+CZ-604D	定価 ¥592,800 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(B)セット: CZ-623TN+CZ-605D	定価 ¥613,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
(C)セット: CZ-623TN+CZ-613D	定価 ¥633,000 ▶ 特価 (価格はお電話下さい)
12回 40,700 24回 21,500 36回 14,900 48回 11,700 60回 9,800	
(D)セット: CZ-623TN+CU-21HD	定価 ¥646,000 ▶ 特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

X68000シリーズ ~P&Aスペシャルセット=限定誌上販売!!

EXPERT-HD

《今月の限定おすすめセット》



①Aセット: P&A厳選セット
■CZ-612C(ブラック)
(本体価格 ¥466,000)

②Bセット: ■CZ-604D(ブラック)
●モニター: スピーカー2個
チルト台付
(定価 ¥94,800)

P & A 超特価 ¥340,000

③Cセット: ■CZ-612+CZ-605D
定価 ¥581,000... ▶ 特価 ¥359,000

④Dセット: ■CZ-612C+CZ-613D
定価 ¥601,000... ▶ 特価 ¥372,000

⑤Eセット: ■CZ-612C+CU-21HD
定価 ¥614,000... ▶ 特価 ¥386,000

■モデム 限定

●PV-A24VM5 (アイワ)

●MNPクラス5
定価 ¥44,800

特価 ¥29,900
(送料・消費税込 ¥31,827)

■40M・外付HDD

●WD-40 (ウィンテック)

●SASI ケーブル付 限定

定価 ¥102,000
特価 ¥58,000
(送料・消費税込 ¥60,770)



回〜84回払いまでOK!!

★頭金なし!★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

X68000用ソフトコーナー (送料1ヶ〜5ヶまで¥500)

Z's STAFF PRO68K Ver2.0 (ソフト)	定価 ¥58,000	特価 ¥39,500
Z's TRIPHONY デジタルクラフト (ソフト)	定価 ¥39,800	特価 ¥27,500
テラツォ (ハミングバード)	定価 ¥19,400	特価 ¥15,840
KAMIKAZE (サムシング・グッド)	定価 ¥68,800	特価 ¥45,500
O & Professional Pack (マイクロウェアジャパン)	定価 ¥58,000	特価 ¥43,000
Final Ver.3.2 (エースビー)	定価 ¥38,000	特価 ¥30,400
C-computer PRO68K Ver2.0 Z-245L	定価 ¥44,800	特価 ¥34,500
CARD PRO68K C2226BS	定価 ¥29,800	特価 TEL下
C-computer PRO68K C2211LS	定価 ¥39,800	特価 ¥28,500
OS-9/X68000 C2219SS	定価 ¥29,800	特価 ¥20,700
AI-68K C2234LS	定価 ¥188,000	特価 TEL下
THE 確率 V2.0 C2224LS	定価 ¥9,900	特価 ¥7,400
SOUND PRO68K	定価 ¥15,800	特価 ¥11,300
MUSIC PRO68K C2213MS	定価 ¥15,800	特価 ¥13,300
Sampling PRO68K C2215MS	定価 ¥17,800	特価 ¥12,500
MUSIC-studio PRO68K 237MS	定価 ¥15,800	特価 TEL下
MUSIC-PRO68K (MIDI) 247MS	定価 ¥28,800	特価 ¥20,500
New-print Shop 221HS	定価 ¥19,800	特価 TEL下
Communication 223CS	定価 ¥19,800	特価 ¥15,500
Communication Ver2.0 C2-257OS	定価 ¥98,000	特価 ¥69,800
TRACE68 Ver3.0 (キャスト)	定価 ¥98,000	特価 TEL下
マイクロンEXPRESS α68	定価 ¥98,000	特価 ¥17,600
68K Ver2 PRO	定価 ¥22,000	特価 ¥14,200
THE FILE PROFESSOR (ロゴシステム)	定価 ¥28,000	特価 ¥22,400
Qツール (サインソフト)	定価 ¥17,800	特価 ¥14,200
たーみのる2 (SPS)	定価 ¥19,800	特価 ¥15,800
マジックパレット (ミカエルプラン)	定価 ¥39,800	特価 ¥24,800
Hyper word C2-251BS	定価 ¥39,800	特価 ¥24,800

●ゲームソフト 20%OFF OK! (一部ソフト除く)

周辺機器コーナー (送料¥500)

A CZ-8NSI	定価 ¥188,000	特価 ¥145,000
B CZ-6VTI	定価 ¥69,800	特価 ¥54,000
C CZ-6TU	定価 ¥33,100	特価 ¥25,000
D BF-68PRO	定価 ¥19,800	特価 ¥15,500
E CZ-6BEI	定価 ¥35,000	特価 ¥26,500
F CZ-6BEIA	定価 ¥38,000	特価 ¥28,600
G CZ-6BE2	定価 ¥79,800	特価 ¥60,000
H CZ-6BE4	定価 ¥138,000	特価 ¥107,000
I CZ-6BF1	定価 ¥49,800	特価 ¥38,200
J CZ-6BP1	定価 ¥79,800	特価 ¥61,000
K CZ-6BM1	定価 ¥26,800	特価 ¥20,300
L CZ-6EB1	定価 ¥88,000	特価 ¥67,500
MAN-S100	定価 ¥36,600	特価 ¥28,500
N CZ-6SD1	定価 ¥44,800	特価 ¥35,000
O CZ-8PC3	定価 ¥65,800	
P CZ-8PC4	定価 ¥99,800	
Q CZ-8PG1	定価 ¥130,000	
R CZ-8PG2	定価 ¥160,000	
S CZ-8PK10	定価 ¥97,800	
T CZ-6PVI	定価 ¥198,000	特価 ¥153,000
U CZ-735X	定価 ¥248,000	特価 ¥190,000
V CZ-8BS1	定価 ¥23,800	特価 ¥19,000

W PIO-6BE1-A (I/O DATA) 定価 ¥25,000 特価 ¥18,000
X PIO-6BE2-2M (I/O DATA) 定価 ¥50,000 特価 ¥36,500
Y PIO-6BE4-4M (I/O DATA) 定価 ¥88,000 特価 ¥64,300

中古パソコンはP&Aにお任せ!!

その場で高価現金買取・高価下取りOK!!

- まずはお電話下さい。 ■下取り・買取でお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。
- 下取りの場合..... 価格は常に変動していますので査定額をお電話で確認して下さい。
(差額は、P&A超低金利クレジットをご利用下さい。)
- 買取の場合..... 現品が着き次第、2日以内に買取額を連絡し、振込み、又は書留でお送り致します。
- 近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

- 月々¥1,000円からOK!! ●ボーナス払いOK (夏冬10回までOK)
●支払い回数 1回〜84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全

全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブル等。
万が一初期不良、輸送トラブルが発生した際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

- マイコン
- ビデオ
- ビデオテープ

P&A

株式会社ピー・アンド・エー

〒124 東京都葛飾区新小岩2丁目1番地19号

☎03-651-0148 (代) FAX 03-651-0141

営業時間

平日: AM10:00〜PM7:00
日祭: AM10:00〜PM6:00

X68000用ハードディスク (送料¥1,000)

アイテム

- HxD-040 (40MB/23ms)..... 定価 ¥118,000 ▶ 特価 ¥88,000
- HxD-042 (増設用)..... 定価 ¥128,000 ▶ 特価 ¥95,000

アイテック

- ITX-640 (40MB/28ms)..... 定価 ¥158,000 ▶ 特価 ¥83,000
- ITX-680 (80MB/20ms)..... 定価 ¥198,000 ▶ 特価 ¥103,000

プリンター (ケーブル・用紙付) 限定5台 新品 (送料¥1,000)

- CZ-8PC3 (カラー漢字24ドット熱転写プリンター)
定価 ¥65,800..... 特価 ¥39,800
- CZ-8PK8 (24ピン漢字プリンター136桁)
定価 ¥152,000..... 特価 ¥49,800
- CZ-8PC4 P&A特選!! (カラー漢字48ドット熱転写プリンター)
定価 ¥99,800..... 特価 ¥58,000

モデムコーナー (送料¥1,000)

- (A) MD-24FS5 (オムロン)..... 定価 ¥49,800 ▶ 特価 ¥34,800
- (B) MD-24FS7 (オムロン)..... 定価 ¥64,800 ▶ 特価 ¥45,000
- (C) コムスター2424/4 (NEC)..... 定価 ¥38,800 ▶ 特価 ¥28,000
- (D) コムスター2424/5 (NEC)..... 定価 ¥44,800 ▶ 特価 ¥32,000

P & A 特選パソコンラック (送料無料) 移動自由 (キャスト付)

③ 3段	④ 4段	⑤ 5段
875 (H) × 580 (D) × 610 (W)	1320 (H) × 600 (D) × 630 (W)	1280 (H) × 600 (D) × 620 (W)
¥9,000	¥11,500	¥15,000

中古パソコン (セットはモニター付) 送料¥2,000

- X68000セット..... ¥180,000
- X68000 ACEセット..... ¥200,000
- X68000 ACE-HDセット..... ¥215,000
- EXPERTセット..... ¥230,000
- EXPERT-HDセット..... ¥265,000
- PROセット..... ¥250,000
- X68000 PRO-HDセット..... ¥270,000
- EXPERT IIセット..... ¥250,000
- EXPERT II-HDセット..... ¥320,000
- PRO IIセット..... ¥240,000
- PRO II-HDセット..... ¥310,000

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕

●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕

●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様の住所・お名前・商品名等をお知らせください。

(電話扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 株ビー・アンド・エー

〔クレジットでお申し込みの方〕

●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。

●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。

●1回〜84回払いまで出来ます。但し、1回のお支払額は¥1000円以上。

超低金利クレジット率

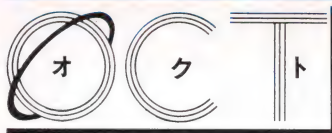
回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.0	4.0	5.5	5.5	10.0	11.5	16.0	21.0	27.0	35.0	42.0



●現金書留及び銀行振込でお申し込みの方は、上記商品の料金の3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合わせ下さい。

朗報です。冬のボーナス一括払い(12月末)OK!!手数料なし!!(1月末払いもOK!!)ご利用下さい。

パソコンプラザ



'90 オクトで始まるパソコンワールド

03-730-6271

●営業時間 AM 11:00 ~ 9:00/日曜・祭日 PM 7:00 電話一本で、ハイ即納
〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

●定休日毎週火曜日 祭日の場合翌日になります。

全国通販

オクト
ラクラククレジット

1回	2.06	3回	3	6回	4	10回	5.5	12回	5.5	15回	8	18回	10
20回	11	24回	12	30回	16	36回	17	48回	22	60回	28		

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回~60回)頭金ナシOK!
- ▶ボーナス一括払いOK!ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト
セレクトデッドシステム

広告掲載商品以外の
製品も取扱っております。



オクト-1

蒲田

- 冬のボーナス一括払い(12月末)OK!!
- 手数料なし!! 平成 1月末払いもOK!!
- ▶今月のセットは、超お買得!! 電話で交渉すべし!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚②ジョイカード 2個(連射式)③シリコンキーボードカバー ④ゲームソフト サンダーブレード(¥9500)

お好みのセットをお選び下さい。

- SX-WINDOW搭載。
- 40Mバイトハードディスク搭載

送料無料



EXPERT II・EXPERT II-HD

- CZ-603C-BK/GY 定価 ¥ 338,000
- CZ-613C-BK/GY 定価 ¥ 448,000

現金特価!! 推選
お電話下さい。

- SX-WINDOW搭載。
- 拡張I/Oポート4スロット装備



PRO II・PRO II-HD

- CZ-653C-BK/GY 定価 ¥ 285,000
- CZ-663C-BK/GY 定価 ¥ 395,000

CZ-8NJ2 限定

●インテリジェントコントローラ
定価 ¥ 23,800
超特価 **¥18,600**



15型カラーディスプレイTV



CZ-605D-GY/BK
定価 ¥ 115,000

15型カラーディスプレイTV



CZ-613D-GY/BK
定価 ¥ 135,000

14型カラーディスプレイ



CZ-604D-GY/BK
定価 ¥ 94,800

21型カラーディスプレイ



CU-21HD
定価 ¥ 148,000

A CZ-603C + CZ-605D.....定価合計 ¥ 453,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

B CZ-613C + CZ-605D.....定価合計 ¥ 563,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

C CZ-653C + CZ-605D.....定価合計 ¥ 400,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

D CZ-663C + CZ-605D.....定価合計 ¥ 510,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

E CZ-603C + CZ-613D.....定価合計 ¥ 473,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

F CZ-613C + CZ-613D.....定価合計 ¥ 583,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

G CZ-653C + CZ-613D.....定価合計 ¥ 420,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

H CZ-663C + CZ-613D.....定価合計 ¥ 530,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

I CZ-603C + CZ-604D.....定価合計 ¥ 429,800▶オクト大特価

12回	¥ 28,000	24回	¥ 14,800	36回	¥ 10,200	48回	¥ 8,000
-----	----------	-----	----------	-----	----------	-----	---------

J CZ-613C + CZ-604D.....定価合計 ¥ 542,000▶オクト大特価

12回	¥ 36,000	24回	¥ 19,000	36回	¥ 13,100	48回	¥ 10,200
-----	----------	-----	----------	-----	----------	-----	----------

K CZ-653C + CZ-604D.....定価合計 ¥ 379,800▶オクト大特価

12回	¥ 25,400	24回	¥ 13,400	36回	¥ 9,300	48回	¥ 7,200
-----	----------	-----	----------	-----	---------	-----	---------

L CZ-663C + CZ-604D.....定価合計 ¥ 489,800▶オクト大特価

12回	¥ 32,200	24回	¥ 17,000	36回	¥ 11,800	48回	¥ 9,200
-----	----------	-----	----------	-----	----------	-----	---------

M CZ-603C + CU-21HD.....定価合計 ¥ 486,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

N CZ-613C + CU-21HD.....定価合計 ¥ 596,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

O CZ-653C + CU-21HD.....定価合計 ¥ 433,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

P CZ-663C + CU-21HD.....定価合計 ¥ 543,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

♡ 本体セットは、11/15~12/14 1ヶ月間だけの大特価セール!!

♡ クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回~60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット: 送料無料 (注) 本体セット以外の周辺機器(プリンター、モデム、HDD等)及びソフトの送料は、北海道・九州地区=1キロ¥1500、■その他離島地区は、1キロ¥2000となります。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

便利です。夜9時まで営業しております。お立寄り下さい。お待ちしております!!

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

■平成2年冬のボーナス一括払いOK!!(12月1月末)手数料ナシ!! 超低金利クレジットをご利用下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

チャンスノ X68000・SUPER-HD(チタン)= 好評・発売中

どんどんTEL下さいネ。

送料ナシ!!

SX-WINDOW搭載。



●ザ・ワークステーションと呼ぶにふさわしい
スーパーな68000!! 新登場!!
SUPER-HD。

※プレゼントノ①MD-2HD10枚 ③ジョイカード(連射式)
②サンダーブレード(¥9,500) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD

●CZ-623C-TN+CZ-613D-TN

定価合計¥633,000...大特価!! TEL下さい。

※マウス・トラックボール付!! ディスプレイにはスピーカ2個、チルト台付!!

12回 ? 24回 ? 36回 ? 48回 ?

♡安くてゴメンなさい。今だけヨ!!

他のディスプレイ(1 CZ-602D、2 612D、3 CZ-603D、
4 CU-21HDの組合せもございますのでお問い合わせ下さい。

※超低金利クレジットご利用下さい。1回~60回払い、頭金ナシノボーナス1回払い、ボーナス2回払いOK!!

X68000 EXPERT-HD

オクト限定スペシャルセット



ラストチャンス!!
早い者勝ち!!

●CZ-612C(BK)
(¥466,000)
●CZ-602D(BK)
(¥99,800)
●MD-2HD 10枚
●ジョイカード(連射式×2個)
●シリコンキーボードカバー

オクト超特価
¥364,000(送料・消費税込み!!)

※ディスプレイ=①CZ-604D ②CZ-605D
③CZ-613D ④CU-21HD
との組合せもございます。TEL下さい。

オクト特選 シャープ周辺機器 (送料¥1,000)

●CZ-6BE1 IBM増設RAMボード.....(¥35,000)▶**特価¥26,500**
●CZ-6BE1B 1MB増設RAMボード.....(¥28,000)▶**特価¥21,000**
●CZ-6BE2 2MB増設RAMボード.....(¥79,800)▶**特価¥60,500**
●CZ-6BE4 4MB増設RAMボード.....(¥138,000)▶**特価¥104,800**
●CZ-6BF1 増設用RS-232Cボード.....(¥49,800)▶**特価¥38,500**
●CZ-6BG1 GP-IBボード.....(¥59,800)▶**特価¥45,000**
●CZ-6BM1 MIDIボード.....(¥26,800)▶**特価¥20,500**
●CZ-6BN1 スキャナ用ハラルボード.....(¥29,800)▶**特価¥22,800**
●CZ-6BP1 数値演算プロセッサボード.....(¥79,800)▶**特価¥60,500**
●CZ-6BO1 ユニバーサルI/Oボード.....(¥38,800)▶**特価¥30,500**
●CZ-6EB1/BK 拡張I/Oボックス.....(¥88,000)▶**特価¥66,000**
●CZ-6VT1/BK カラーイメージ・ユニット.....(¥69,800)▶**特価¥53,000**
●CZ-6BL2 LANボード.....(¥298,000)▶**大特価**

●CZ-8NM2A マウス.....(¥68,800)▶**特価¥5,300**
●CZ-8NT1 マウス・トラックボール.....(¥98,800)▶**特価¥7,500**
●CZ-8NS1 カラーイメージスキャナ.....(¥188,000)▶**大特価**
●CZ-6BC1 FAXボード.....(¥79,800)▶**特価¥60,500**
●CZ-8TM2 モデムユニット.....(¥49,800)▶**特価¥38,000**
●CZ-64H 増設ハードディスク.....(¥120,000)▶**大特価**
●CZ-6TU GY/BK RGBシステムチューナー.....(¥33,100)▶**特価¥25,000**
●BF-68PRO 高性能CRTフィルター.....(¥19,800)▶**特価¥15,500**
●SX-68M(システムサブM) MIDIボード.....(¥19,800)▶**特価¥15,000**
●PIO-68BE1A(I/O DATA) 1MB増設RAMボード.....(¥25,000)▶**特価¥18,000**
●PIO-68E2-2M(I/O DATA) 2MB増設RAMボード.....(¥50,000)▶**特価¥36,500**
●PIO-68E4-4M(I/O DATA) 3MB増設RAMボード.....(¥88,000)▶**特価¥64,300**
●CZ-6BV1 ビデオボード.....(¥21,000)▶**特価¥15,800**

オクト面白グッズ

アイテック(送料¥1,000)

●IT-X640(¥158,000)
.....**特価¥103,000**
●IT-X680(¥198,000)
.....**特価¥134,000**

モデムコーナー(送料¥1,000)

●MD-1200A III.....**特価¥14,800**
●MD-24FS4.....**特価¥31,500**
●MD-24FS5.....**特価¥34,800**
●MD-24FP4.....**特価¥27,900**
●MD-12FS.....**特価¥15,000**

熱転写カラー漢字プリンター (ケーブル付) 送料¥1,000



CZ-8PK10

24ピン
カラー漢字プリンター136桁

①CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800.....**大特価!! TEL下さい。**
②CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000.....**大特価!! TEL下さい。**
③CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000.....**大特価!! TEL下さい。**
④IO-735×(カラーイメージジェット)
定価¥248,000.....**大特価!! TEL下さい。**

パソコンラック 推奨 無料

①五段キャスター付 ②四段キャスター付 ③三段キャスター付



5段キャスター付
キーボードが収納できる
から、手元でマウス操作が
ラクができる
棚板5段のマルチに
活用できるデスク。
ウーン、こいつはデキルノ!
(1325(H)×640(W)
×700(D))

特価¥16,000



4段キャスター付
どんなパソコンにも
フレキシブルに対応!
使い易いデスクです。
(1245(H)×614(W)
×600(D))

特価¥12,000



3段キャスター付
場所を選ばない
簡易で便利な
デスクです。
(1175(H)×640(W)
×600(D))

特価¥8,800

X68000ソフト大セール実施中※ゲームソフトオール25%off 送料¥500

クラシック	型名	商品	定価	特価
●Z's STAFF PRO68K Ver.2.0 (ソフト)定価¥58,000	CZ-211LS	Complier PRO-68K	¥39,800	¥28,800
オクト特価¥40,000	CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
●データベース ●KAMIKAZE (サムシングソフト)定価¥68,000	CZ-213MS	MUSIC PRO-68K	¥18,800	¥13,500
オクト特価¥46,000	CZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
●グラフィック ●C-TRACE68 (キャスト)定価¥68,000	CZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
オクト特価¥51,000	CZ-219SS	OS-9 X68000	¥29,800	¥21,000
●C言語 ●C & Professional Pack (マイクロウェアジャパン)定価¥58,000	CZ-220BS	DATA PRO-68K	¥58,000	¥41,000
オクト特価¥44,000	CZ-257CS	Print Shop PRO68KV.2	¥19,800	¥14,300
●グラフィック ●サイクソン エキスプレス 定価¥78,000	CZ-223CS	Communication PRO-68K	¥19,800	¥14,300
オクト特価¥58,000	CZ-224LS	THE 複製 V2.0	¥9,900	¥7,500
●グラフィック ●デジタルクラフト 定価¥39,800	CZ-226BS	CARD PRO-68K	¥29,800	¥21,300
オクト特価¥28,000	CZ-241BS	システム手帳リファレンス集	¥9,800	¥7,500
●ワープロ ●ハイパーワード 定価¥39,800 CZ-251BS	CZ-242BS	活用フォーム集	¥9,800	¥7,500
オクト特価¥29,800	CZ-244SS	Homan 68K Ver.2.0	¥9,800	¥7,500
	CZ-247MS	MUSIC PRO-68K(MIDI)	¥28,800	¥20,800
	CZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
	CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
	EW		¥38,000	¥29,800
	G-68K		¥14,800	¥11,400
	E-68		¥19,800	¥15,300

★オクト今月だけの新品限定販売(各1台限)(送料¥1,000)

●CZ-611C(BK) 定価¥399,800.....**大特価¥218,000**
●CZ-652C(BK) 定価¥298,000.....**大特価¥188,000**
●CZ-662C(BK) 定価¥408,000.....**大特価¥248,000**
●CZ-601D(BK) 定価¥119,800.....**大特価¥68,000**
●CZ-601D(GY) 定価¥119,800.....**大特価¥68,000**
●CZ-612D(GY) 定価¥119,800.....**大特価¥74,000**
●IO-735 定価¥248,000.....**大特価¥158,000**

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-730-6271

お申込みはお電話でお願いします。お客様の(住所)氏名(電話番号)及び(商品名)をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い

銀行振込:お近くの銀行より(電信扱い)にて
お振込み下さい
現金書留:封筒の中に住所・氏名・商品名を
ご記入の上当社までお送り下さい

クレジット

専用お申込用紙をお送り致します
ので、必要事項をご記入、ご捺印の上
ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2.0%	3回	3%	6回	4%	10回	5.5%	12回	5.5%
15回	8%	18回	10%	20回	11%	24回	12%	30回	16%
36回	17%	48回	22%	60回	28%				

振込先

富士銀行 三菱銀行
久ヶ原支店 蒲田支店
①No.1824 ②No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。

※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。



The スーパー

お待ちどうさま。いよいよ登場です。



創刊記念
ごほうびあげちゃう

スーパーファミコンソフト
150本
プレゼント

特別付録

ファイナル
ファイト読本

スーパーファミコンまるかじり!

隔週金曜日発売 定価380円(税込)

ファミコン

11月16日創刊

3大特集

新作ソフトでみる!
これがスーパーファミコンだ

性能でみる!
これがスーパーファミコンだ

データでみる!
これがスーパーファミコンだ

スーパーマリオワールド

F-ZERO

グラディウスIII

ポピュラス

SDザ・グレイトバトル

その他新作情報満載

すぎやまこういちのゲーム漂流記
第1回ゲスト

いとうせいこう

「ザ・スーパーファミコン」

創刊2号は11月30日(金)発売!!

もちろん特別付録もついちゃって…乞うご期待!

■最寄りの書店でお早めにお求め下さい

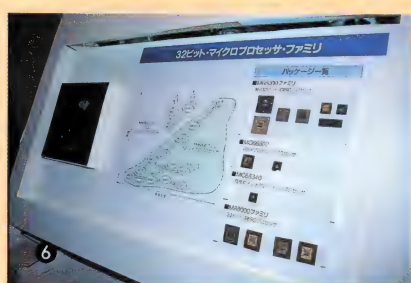
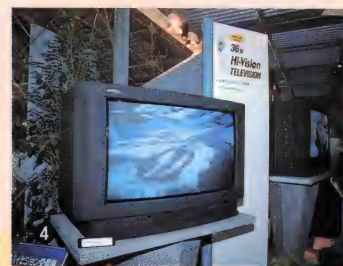
ソフトバンク出版事業部

東京都港区高輪2-19-13 電話03(5488)1360

エレクトロニクスショー&データショー'90

10月、東京晴海の国際見本市会場において2つのショウが開催された。2日から6日までがエレクトロニクスショウ'90、そして10月22日から25日までがデータショー'90。この模様を写真を中心に紹介していきたい。

Japan Electronics Show エレクトロニクスショー'90



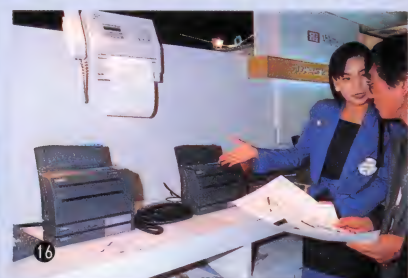
- ①エレクトロニクスショウ会場風景
- ②明るくともよく見えるシャープの高輝度プロジェクタ
- ③パナソニックの高速応答液晶モジュール
- ④ハイビジョンはほとんどのメーカーが36型を展示
- ⑤カシオはアウトドアな雰囲気液晶テレビをアピール
- ⑥MC68340を含むMC68000ファミリが勢揃い
- ⑦CD-Iエレショウ推進協議会のブース
- ⑧エレショウのイメージキャラクタ? のオブジェ
- ⑨富士通は新型ASPをスーパーモナコGPでデモンストレーション
- ⑩鈴木亜久里選手でおなじみ、DynaBookの東芝はF1を



データショー'90



- ⑪データショウ会場風景
- ⑫あのMac II (カラー版)が318,000円! SE相当も10万円台に!
- ⑬巨大なハイパー電子手帳
- ⑭いかにもプリンタのエプソンという感じ
- ⑮鮮やかな色を見せるTFTカラー液晶搭載のAX386LC
- ⑯富士ゼロックスのプリンタ兼コピー機、FAXにもなる



同人ソフトばかりがアマチュアプログラムじゃないぞ。1987年5月の開発開始から何度かの中断を経て、いまだ未完成。でも、自分の趣味で作るからここまでこだわれる。本当の“USER'S WORK”を紹介しよう。

今月はちょっと趣向を変えてX1turbo専用の「グラディウス」を紹介する。画面を見れば一目瞭然、あのコナミのグラディウスだ。驚かれる方もいるだろう。すでにX1版のグラディウスはコナミから発売されている。でも、作ってしまったものはしょうがない。作者は東京都の横内威至君(17)だ。たったひとりでX1turbo版グラディウスを作り上げた。

とにかく凄い！

最初は投稿作品として送られてきたものだが、残念ながら著作権問題を抱え、掲載できる大きさでもない(ディスク2枚組)。しかし、その完成度たるや尋常ではない。できるだけ多く画面を載せたのでじっくり見てほしい。デジタル8色のみでここまでグラフィックを再現した力量は驚嘆に価する。止め絵だけでなく、アニメーションパターンもかなり本物に近い。爆発パターンまで細かい。諸々の事情によりプログラムの配布ができないのが非常に残念である。

凄いのはグラフィックばかりではない。いや、逆にこれだけのグラフィックなら動きが重そうだと考える人もいるかもしれない。X68000を除いて、これまでパソコン用として発売されていたグラディウスは、処理速度の関係からかオプションの数を制限していたくらいなのだから。

しかし、編集室に届いたテスト版では、お遊びといいながらもオプションを最大12個つ



けてみせる。これですべてのオプションからレーザー、ミサイルを発射してもちゃんとゲームできる速度である。さほど重くはならない。さらに背景にはちゃんと3重スクロールの星が瞬きながら流れている。自作の音源ドライバからはFM8声のBGMに、PSGの効果音だ。

動きその他にも、これまでX1用に市販されてきたシューティングゲームのなかでも、はたしてこれ以上のゲームがあったらどうか？と思うくらい凄い。ゼビウス、サンダーフォース、スーパーレイドック、ルクソールなどよくできたシューティングゲームと比べてもまったく遜色ない。

ここに画面写真で挙げたものはまだ未完成なのだが、なにより凄いのは細部へのこだわ



原作に忠実なグラフィック。この画面がしっかり動きます。本当に「グラフィックだけではありません」。本当に動きも凄いいけどなあ……。

りだ。

各面が終了すると背景はゆっくりとフェードアウトする。最終面をクリアすると敵要塞が縮小表示されていき、ビックバイパーが飛び出すところまで作ってある。さらにはX68000でも省略されたアーケード版と同じネームエントリーが用意されている。

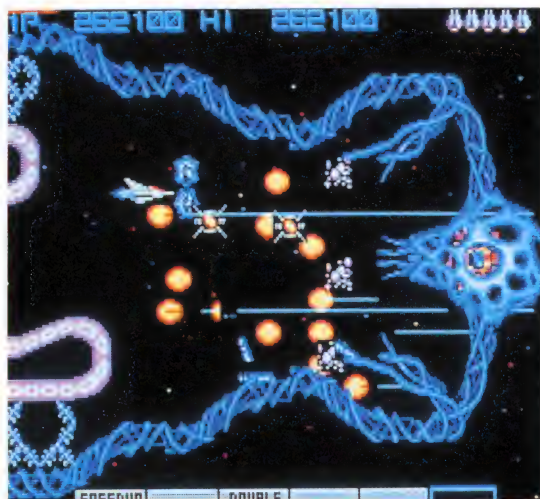
X68000版との比較という無謀な試みをして、あえて欠点を挙げると、

キャラクターがやや小さい
ザコキャラの動きが粗い

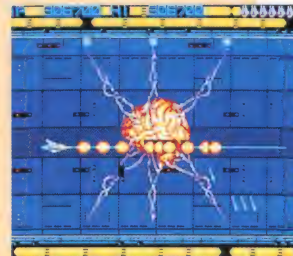
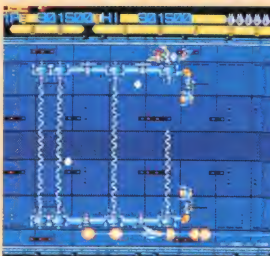
2, 3面などで縦スクロールしない
スクロールが8ドット単位

一部のキャラで重ね合わせをしていない
といったところ。8ビット機だということとPCGの制約を考えればしかたないことではある。

X1turboでもこれだけのことができることを証明した1作。X1turboにも不可能はないのか？



まだ完成していないのだが、各ステージの様子を見てみよう。オプションの数はご愛敬として、モアイステージは未完成、触手ステージはまだない。しかし、これがデジタル8色のグラフィックとは……。



愛読者プレゼント

おかげさまでOh! Xも誌名変更3周年、ホントにホントにありがとう。さて、今年も各方面からたくさんの提供品をいただいたので、今月と来月2回にわたってプレゼントです。まず今月は我々がシャープ編。応募方法は185ページを見てね。

X68000オリジナルグッズ



1
2名

X68000シースルークロック

6,800円

以前にもアナログ時計はあったが、こちらはデジタルタイプの置き時計。もちろんアラーム機能もついている。



3
5名

X68000ジッポ・ライター

4,800円

ご存じX68000のイメージキャラクター (?), ツタンカーメンの絵入りオイルライター。持ったときの感触がなかなかよい。

2
2名



X68000クリスタルポルシェ

8,000円

人気の高いスポーツタイプの車を象った置物。なかなかキレイな代物だ。見・体・験フェアなどでも人気の商品。



4
5名

X68000キーホルダー

1,300円

これをさりげなくカバンにつけていると「おお、キミもX68000ユーザーか」と街中で突然仲間ができるかもしれないぞ (ホントか?)。

オリジナルソフト

5 5名



X68000ネクタイピン

3,000円

風に遊んでいるネクタイもいいけど、やっぱりオトコならひとつぐらいネクタイピンを持っていたい、そんなキミにあげたい品だ。

6 2名



X68000ポーチ

4,000円

ちょっと出掛けるときには便利なこのポーチ。外側にチャックもついていて、使いやすいことこのうえない。

7 2名



X68000ボストンバッグ

4,800円

旅行に出掛けるときにはこれ、ボストンバッグ。肩に掛けられるようにちゃんとベルトがついているのもうれしい。

8 1名

XBAS to C CHECKER

X68000用 5"2HD版
9,800円

X-BASICで書かれたプログラムをコンパイルしたときに出るエラーもこれで解消。これひとつでプログラムの修正もラクになるぞ。

9 1名

X68000用
5"2HD版
44,800円

バージョンアップしたC compiler PRO-68K。ソースデバッカをはじめとしてソフト開発には欠かせない機能を満載したソフトだ。



10 1名

CANVAS PRO-68K

X68000用 5"2HD版
29,800円

ドロー系グラフィックツールとして注目のこのツール。細かな曲線や、拡大縮小機能、変形もできる。印刷の美しさも見逃せない。



熱血高校ドッジボール部 サッカー編

11 3名

X68000用
5"2HD版2枚組
8,800円

もうお馴染みのくにおくんシリーズ。今回は助っ人でサッカーに挑戦。もちろんサッカーのルールなんて知らなくても楽しめる。



SOFTWARE information

今年も年末年始にかけて発売されるゲームがもりだくさん。移植ものあり、オリジナルものあり、“2”ものありと、バラエティも豊かです。これで冬休みは退屈しないですむかな？

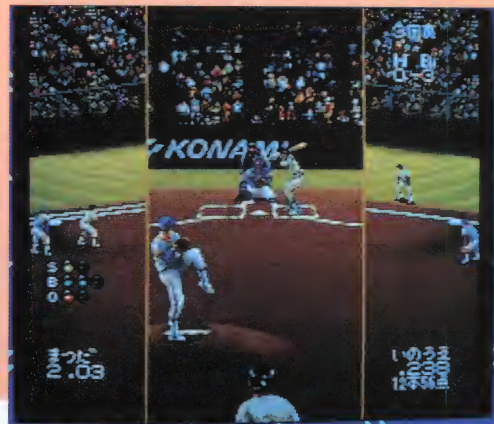
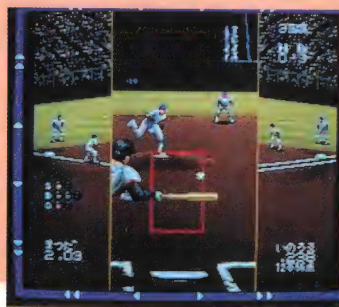


画面は開発中のものです

パロディウスだ！
お馴染みグラディウスのパロディ版、パロディウスだ！の登場。デフォルメされた動物たちがなんとも可愛い。

生中継68

コナミのX68000オリジナル野球ゲーム。野球中継風に投げるとき、打つときの画面が見られるのも面白い。



話題のソフトウェア

ふ、ふあつくしよん、っとー、あー、カゼひいたあ。あ、いきなりドーモすいません、皆さんはお元気ですか？突然寒くなったもんで、デリケートなあたしは気候の変化についていけなかったの。やっぱり美人薄命なのね……。くしゅんっ。んなカゼくらいでオーバーなって？いいじゃない、一度いってみたかったんだから、このセリフ。

いやー、今月もゲームが多いこと多いこと。もう各ソフトハウスさんの年末にかけて意気込みが感じられますね。それじゃ、さっそくいってみましょー！

今月トップを飾るのはコナミ。2本一挙に発表です。まずは皆さんの期待に応えるべく登場の**パロディウスだ！**今年アーケードで人気だったこのシューティングゲームが、いよいよX68000に移植です。プレイヤーは、おちゃめで可愛いマイキャラ（ビックパイパー、たこ、ツインビー、ペン太郎）を操り、これまた可愛い敵キャラを倒していくというもの。もちろん、この

やっぱりズームは強かったのだ

- | | |
|--------------------|------|
| 1 ラグーン | 2 ↑ |
| 2 シムシティー | 1 ↓ |
| 3 サイバリオン | 10 ↑ |
| 4 ソーサリアン（含追加シナリオ） | 7 ↑ |
| 5 ポピュラス（含プロミストランド） | 4 ↓ |
| 6 ダンジョン・マスター | — |
| 7 エアー・コンバット〜遊撃王II〜 | —初 |
| 8 パロディウスだ！ | —初 |
| 9 機甲師団 | —初 |
| 10 ワンダラーズ・フロム・イース | 5 ↓ |

ラグーンがシムシティーを押さえてトップに躍り出ました。どちらも発売前から票を集めていただけに発売後の動向に興味津々でしたが、X68000ユーザーはやはりアクション要素のあるものが好きなようです。

しかしラグーンは、支持が「すべていい！」という熱狂的な人と、「次回への期待票の意味を込めて」というやや冷めた目で見ている人に分かれています。どちらもズームのセンスには高い評価を下しているようですが、トップを維持できるかどうかはまだ流動的。

10位からジャンプアップして3位につけたのがサイバリオン。原作の独自性、移植の完成度について言及しているハガキが多いのですが、なかにはMIDIに対応していないのが不満という声も。そろそろMIDIも必須アイテムになりつつあるということでしょうか。

6位にはダンジョンマスターがカムバック。先月ポピュラスが4位でがんばっているのにランク外になったのがファンにとっては、心外だったらしく、「まだまだ遊んでます」というハガキがどっきりきました。

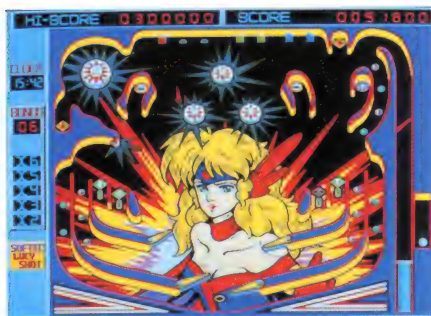


ワールドスタジアム

ゲームを知らない人でも察しがつくとおり、これはグラディウスのパロディ版。でも、そんなこと関係なしにこの楽しい雰囲気を味わってほしい、そんな作品なのです。発売まで楽しみに。

で、もう1本はX68000オリジナル野球ゲーム生中継68。アニメーションやカメラアングルで、あたかもTVの野球中継をそのまま見ているような気にさせてくれるゲームです。エディット機能で細かいところまで設定変更でき、自分好みのチームを作れるのもうれしいところ。こちらは1月発売予定。もうちょっと待ってね。

続けてもう1本野球ゲームを紹介しちゃいましょう。こちらはナムコのアーケード版からSPSが移植した、ワールドスタジアムです。これはゲーセンで遊んだ人もかな



LUCY・SHOT

7～9位はすべて初登場。ハガキの声は……。

エアー・コンバット～遊撃王II～：サイバースティックを使うと気分が出る。操作がシンプルでなじみやすい。空中戦が面白い。ただスピード感はいまいち。

パロディウスだ！：写真で見ただけが完成度が高いと思った。アーケードの原作が好きなので。グラディウスIIIも期待している。

機甲師団：きれいなグラフィックと難しめのシナリオがいい。もう少し速ければもっとよかった。ウォーシミュレーションとしては異色。

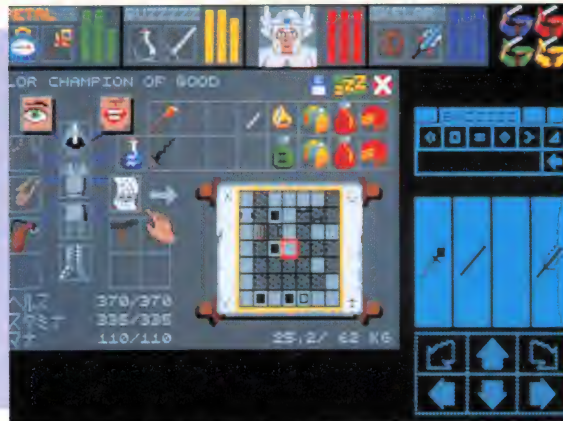
ところでTOP10では「キライだから」というハガキも1票に加えているので、本当にこのソフトはダメだと言いたいときは「編集部へのメッセージ」欄をお願いします。ちょっと不満も言っておきたいが好きだというのは「推薦する市販ソフト」欄でかまいません。では。（浦）



続ダンジョン・マスター

カオスの逆襲

あのダンジョン・マスターの続編。今回はさらに入り組んだダンジョンが君を待つ。エディタで自分の好きな顔を描けるのもいい。



りいるんじゃないでしょうか。移植にあたっては、音声データや効果音はアーケードとほぼ同じ、もちろんオールスター戦でのエディット機能も健在ということなので、期待していてもよさそう。これは要2Mバイトです。12月14日発売の予定。

12月14日発売予定といえば、こちらもそうビクター音楽産業の続ダンジョン・マスター カオスの逆襲。これはタイトルどおり倒したはずのロード・カオスが、なんと生きていた、かくして勇者は再びダンジョンへ……という筋書き。もちろん「ダンジョン・マスター」で大切に育てたキミの勇者をこの続編でも使用できる。ただ、かなり強くなっているという設定上、あまり育っていない勇者だとそうとう苦しいはず。というわけで、もちろん新しい勇者もちゃんと用意されています。今回はダンジョンも難しいためオートマッピング機能もついています。そして新しい魔法や、デモアニ

メーション、それにFM音源にも対応しました。勇者の顔もエディタで描き換えられるので、一層親しみがわきますね。

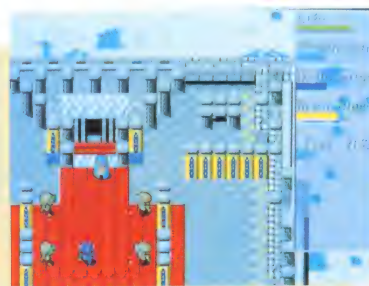
さて、ピンボール・ピンボールを発売したばかりの日本ソフトックでは、なんともう第2弾を発売しちゃいます。早いですねー。今回はピンボールもので、タイトルはLUCY・SHOTです。前回の中国風とはうってかわって、今回はアメリカナイズされたグラフィックが印象的。

そして、ハミングバードのラプラスの魔、これもやっとな姿を現してきたよう。ゲームの内容はホラーRPG、画面写真からオドロドロしい雰囲気が伝わったでしょうか。もう少ししたら詳しく紹介できそうなので、待っていてくださいね。

プラザ工業（タケル）からはハイドライド3 SV（スペシャルバージョン）がすでに発売されています。皆さんもうプレイしてみましたか？ あのARPGの名作が



ラプラスの魔



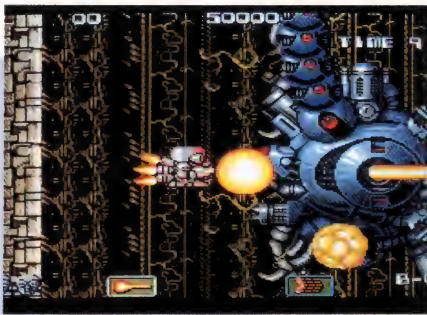
ハイドライド3 SV



ブルトン・レイ



銀河英雄伝説II



アトミック・ロボキッド

かタイヘン。またまたシューティングファンには見逃せない一作となりそうです。12月発売の予定です。

さて、今月3つ目の野球ゲームです。アートディンクでは、**栄冠は君に**を開発中。このゲームは先ほどの2つとは違い、高校野球を題材にしたもの。約4,000校の出場校を相手に、トーナメントを勝ち抜いていくというもの。ユーザー優勝プレゼントに、深紅の大優勝旗も用意されているぞ。

さて、テーブルトークの老舗、ホビージャパンからは**リングマスターII**の登場です。前作から1年たったいまでも固定ファンのいるリングマスター、今回はシステム的にはそう変わりなく、ファンを喜ばせてくれそう。まだ開発中ですが、画面を見ていただければ出来はわかっていただけると

思います。こちらもうちよつとしたら詳しくお伝えするつもりです。

最後に画面は載せられなかったけど、タイトルだけでも紹介していきましょう。

まずはシムシティー以降の動きが気になるイマジニア、こちらは**シムアース**の開発に取り組んでいる様子。発売は来年3月あたりになりそう。

データウエストからは、お馴染みの「Misty」シリーズ、**Misty Vol.6**が発売されます。X68000, X1turbo同時発売で11月22日発売です。なお、タケルでも発売されます。

X68000オリジナルで話題のウルフ・チームでは、シューティングゲーム**ソル・フィース**を発売予定。詳しくは来月。

スタークラフトではPC-9801などで発売されている**ファンタジーIV**をX68000に移植



栄冠は君に

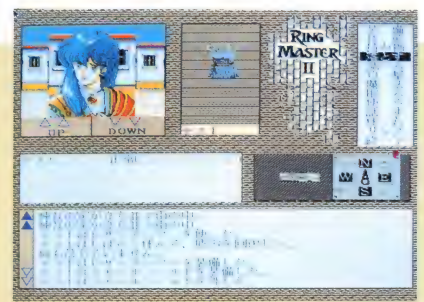
開発中。発売は来年1~2月頃になりそう。

ゲーム以外にも、ツァイトからZ'sTRIPHONY・DIGITAL CRAFTのデータ集**3D倶楽部**が発売になりました。これはお部屋のインテリアパーツをいろいろと用意し、画面上に自分なりの部屋をデザインしようというもの。なかなか本格的ですよ。

そうそう、**C-TRACE68+**もキャストから12月に発売されます。こちらはC-TRACEのバージョンアップ版。198,000円とちょっと高めだけれど、ついにメタボールが使えるぞ。

サン・ミュージカルサービスの**Mu-1**がバージョンアップ。FM音源関係や和音の確認機能など。登録ユーザーには無償バージョンアップされる模様。Musicstudioもバージョンアップされる予定。

それでは、また来月。



リングマスターII

服して、星系を我が物にするのだ!

X68000用 5"2HD版2枚組 12,800円(税別)
工画堂スタジオ ☎03(353)7724

★リングマスターII 永遠なる想い

前作「フィリアス・ノギスの暗雲」の主人公は、リングナイツの座を手に入れて再び旅立った。戦乱を予感した彼は、隣国ムルソン大公国へと向かう。が、そこでも新たな暗雲がたちこめていた……。充実した世界設定とシナリオ、綿密な戦闘システムを持つRPG。人物との会話に重点を置いてあり、テーブルトークの老舗らしい独特の雰囲気を持つゲームである。

X68000用 5"2HD版 価格未定
ホビージャパン ☎03(354)9341

★生中継68

コナミから久々のアクション型野球ゲームが登場。「生中継」というだけあって、X68000のグラフィック機能を活用して、試合の興奮をそのまま伝えることに主眼を置いている。画面のアンギュラはTV中継と同様にし、アニメーションによって臨場感を出す。エディット機能とゲームモードの種類も充実させて、ユーザーの細かい要求にも応えてくれそうだ。

X68000用 5"2HD版 価格未定
コナミ ☎03(262)9110

★ブルトン・レイ

中世ファンタジーの世界に基づいたフィールドタイプのRPG。登場人物すべてが自分の思考を持っている点が特徴で、戦闘もプレイヤー以外のキャラクターは自分で行動を決定する。ゲームは複

数の短編シナリオからなり、レベルアップしたキャラクターやアイテムを、ほかのシナリオに引き継ぐこともできる。色調を抑えたグラフィックや、リュートを意識したBGMなど、中世ヨーロッパの雰囲気を出すことにも気を使った作品だ。

X68000用 5"2HD版2枚組 価格未定
システムソフト ☎092(752)2602

★ラプラスの魔

ラプラスの魔は、原作は安田均氏、音楽に小坂明子氏と豪華なスタッフを迎えて贈るホラーRPG。アメリカはマサチューセッツ州東部の「ウェザートップの館」という洋館の中で、ゴーストハンターと呼ばれる人々と怪物たちの戦いが繰り広げられる。そして、舞台は館を越えてさらに……。精神的なダメージなども考慮され、ホラーRPGらしいデザインに仕上げられている。

X68000用 5"2HD版2枚組 価格未定
ハミングバードソフト ☎06(315)0541

★栄冠は君に

アートディンクの新作は高校野球のシミュレーションゲーム。去年地区予選に参加した全国の高校約4,000校がすべて登場する。プレイヤーは監督となってチームを率い、部員の能力を把握しながら40日間の練習をこなし、地区予選、そして全国大会へと進んでいく。自信や気力などの高校野球らしい要素を加えて監督としての難しさを体験できるゲームだ。それだけに優勝したときに見るデモは感激ひとしお。

X68000用 5"2HD版3枚組 9,500円(税別)
アートディンク ☎0474(77)7541

★アトミック・ロボキッド

UPLのアーケードゲームがシステムサコムによってX68000に移植・発売される。左右スクロール型のアクションゲームだ。プレイヤー操るロボキッドは、人類の存亡の鍵を握るプログラムを持ちながら、その使命を知らないというなかなかハードな設定。全25面で、1画面分以上もあるボスキャラや、同型機との対戦もある。音楽はMIDIにも対応。

X68000用 5"2HD版 8,800円(税別)
システムサコム ☎03(635)7609

★ワールドスタジアム

ファミコン、アーケード、PCエンジンなどで好評を博したナムコの野球ゲーム、「ワールドスタジアム」がX68000にも登場する。12のプロ野球のバロディ球団、3つのスタジアムを選んで対戦する。ヒッティング、バント、牽制球やダブルスチールまで多彩なアクションを使い分けて勝利を目指すそう。2人プレイも可能。要2Mバイト。

X68000用 5"2HD版 8,800円(税別)
SPS ☎0245(45)5777

★C-TRACE68+

キャストのレイトレーシングソフトウェア「C-TRACE」がバージョンアップした。メタボールにより有機的質感が表現でき、αチャンネルへの対応により高度な合成作業を可能とした。光の当て方の設定も自由になり、照射範囲の境界をぼかすこともできる。

X68000用 5"2HD版 198,000円(税別)
キャスト ☎03(705)1065

シューティングゲームの王道である



Nakano Shuichi
中野 修一

あの「R-TYPE」のアイレムが、久々にX68000のゲームを発表。それもアーケードで秀作との呼び声の高い「イメージファイト」だ。もうすぐ発売、それまでウデを鍛えておくではないか。

少し飢えていた、と思う。バリバリのガシガシのシューティングゲームに、だ。硬いボスキャラ、貧弱な武器、複雑な操作、なにがゲームバランスだ、アクションRPGだろうが！ えーい、一方的に撃ってくるんじゃない！ どんどんフラストレーションがたまる。

ということで、イメージファイトだ。この年末にはいくつかシューティングゲームが発売されるが、ガンガン撃ちまくりたいという人にはイチ押しの作品。こと、こういうことをやらせるとX68000のハードウェア構成は強力だということを思いしらせてくれるぞ。うーん、2重スクロールにスプライトはあくまで滑らか。CYNTHIA(X68000のスプライトコントローラ)はいい仕事をしている。

イメージファイトとは

もともとイメージファイトとはアイレムが1988年に発表したシューティングゲーム。あのゲーム史に残る名作を作ったR-TYPE開発部隊の作品とあって注目を浴びた。それがアイレム自身の移植によってこのたびX68000にお目見えとなった。とにかくX68000では久々の縦スクロールシューティングゲームだ。めでたい。

まずプレイヤーはシミュレータでテストを繰り返し、それにパスしなければ実戦には参加できない(……こともないが、補習を受けなければならない)。前半の5ステージがシミュレータで後半の3ステージが実戦だ。要するに1~5がイメージで残りがファイトなわけだ(?)。

特徴的な攻撃方法を解説しよう。

まず、バラ弾を撃つ。ノーマルショットというやつだ。連射がきき、かなり強力。また、自機の変速を変えるとバックファイアが発生し、後方の敵にダメージを与えることができる。これもものちのち便利になることもある。

そしてオプション兵器。ポッドというものがある。自機の左右と後方、最大3機装着できる。青いポッドは前方に弾を撃ち、赤いポッドは進行方向の逆に弾を撃つ。赤いポッドとお友達になることがイメージファイトの醍醐味に通じる。狭い通路でもひよいと左右を撃ち分けたり、どてっ腹に弾を集中したり、クルクルとまわりじゅうに弾をばらまいたり、とにかく頼りになる。間違えて青いポッドを取ってしまったら、左右は体当たりに限る。ポッドシュートといって左右のポッドを前方に飛ばす技もある。

頭部には特殊兵器のユニットが装着できる。派手なリングレーザー、サーチレーザー、誘導ミサイル、ワイドレーザーなど好みで使い分ければよい。これらは1回分のバリアとしても役に立つ。

が、とりあえずポッドさえあればなんとかなる。飛び道具満載のシューティングゲームでも、ゲーム感覚は「近接戦闘」なのが面白いところ。

原作との違い

最初に水を差すようで悪いが、もともとアーケードというのは、とんでもない世界ではある。オーバースペックなハードウェアによってたかってソフトを載せる。パソコンで完全な移植なんてそうそうできるものではない。やはり原作と並べれば見劣りせざるをえない。そこんことを踏まえたくて読んでほしい。

アーケード版では縦画面だったこのゲーム、X68000では横画面に変更されている。よって、一部の場面でアーケード版のゲームバランスが再現されていない。が、ドラゴンスピリットの横画面が気にならない人ならまず問題ない。一部のボス部分だけは



まだボスキャラではありません

バックスクロールで対応しているようだ。

画面はまったくゲーセン版に遜色ない。動きもやや粗いがまあ合格。高速スクロールがときどきガタつくことがある以外、全体の動きはかなり軽い。最初のうちは敵キャラの出現頻度などでもの足りなく思えることもあると思う。やがてそれは相当手加減されていたということに気づくだろう。極端にキャラが増えると途端に重くなるゲームが多いなかで、これだけ余裕を感じられるゲームも珍しい。R-TYPEとはえらい違いだ。

細かい部分で敵の硬さやバランスが変更されているようだが、厳として「イメージファイト」以外の何者でもないと感じさせる仕上がりである。PCエンジン版やファミコン版の忠実度とは比較にならない。

難易度は?

これだけ気合いの入った移植だからゲームも本格的だ。もともとゲーセンの達人ゲ



これがポッドシュートだ!

X68000用 5"2HD版2枚組 8,800円(税別)
アイレム ☎06(535)4880

ーマー用に作られたもの。そのままでは素人には手が出ない。そこで自動連射モード。これでハンデはなくなった。

ファミコン版、PCエンジン版でのHARDモードよりX68000版のEASYモードのほうが難度が高いという噂もあるが（事実だろう）、1周目に限定すれば適正な難易度といえる。ドラゴンスピリットやサンダーフォースIIに比べればま

だ手軽だし、ランクは4段階に設定できる。モードに限らず、敵の攻撃を見切れば、別に「反射神経の人」でなくてもクリアできる。

このゲームでは一概に、

難易度を上げる＝敵が硬くなる

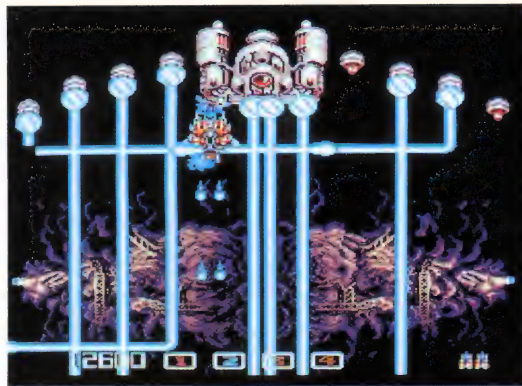
ではなく、ランクを上げて個々の敵の硬さはあまり変わらない。変わるのは敵の出現頻度（と弾の速度）だ。とりあえず進みたいという人はEASYを、たくさん敵が出るほうが得した気になる人はランクを上げたほうがいい。実はVERY HARDでも最高に難しいわけではない。どうせ、EASYでも2周目になれば1周目のVERY HARD以上の難易度になるのだ。

また、ノーコンティニューかどうかでキャラクターの出方も変わる。途中で中ボスひとつ加わるだけでもゲームの緊迫感が違ってくる。さらに、赤いポッドを取ると難易度が上がるという噂もある（未確認）。

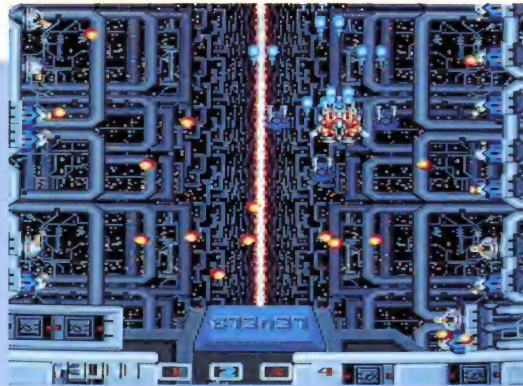
ちなみに、アイレムの伝統かどうかは知らないが3周目はない。2周目をクリアした時点でゲームは終了する。まあ、無限コンティニュー制とはいえ、一般庶民にはあまり縁がない世界ではある。

また、イメージファイトの特徴としてシミュレータでの撃墜率が90%以下なら最終面より難しいといわれる補習ステージをこなさなければならないことがある。

つらい。まずパターンを見切って攻略法を摸索する。と、できあがったプログラム



紙一重の美学。安全地帯がなんでもい！



いよいよ大詰め。最終ステージへ

には一瞬の遅れも許されないということがわかる。敵の出現予定位置に先回りし、密着して連射、敵が爆発を始めるタイミングで次のターゲットに移動。場合によっては、一瞬だけ飛び出して敵の攻撃を誘導しておかなければかわせないこともある。もちろん、スピードは3以上だ。複合攻撃をいかに避けるかがポイントになりそう。

各面をきちんとクリアすれば縁がないが、一度は挑戦してみたいステージだ。

ここがいい！

現れるものをすべて破壊しなければならない。逆にいえば、すべて破壊できる、というコンセプトがいい。敵の攻撃ばかり激しくて、撃ちまくるより逃げまくることが要求されるゲームが多い昨今、こういうストリートな作りが新鮮だ。

R-TYPEにもいえたことだが、フル装備状態で死んでも途端に絶望的になったりはしない。どこからでもちゃんと復活できるのがいい。復活できる。信じればできる。

敵キャラはザコでも大型で、なおかつアニメーションパターンも多い。なによりデザイン、描き込みが美しい。とにかく派手で綺麗なのがいい。

キャラクターのレスポンスがいい。無骨なデザインの自機もいい。左右移動で機体がアニメーションするのは当然として、速

度を変えれば「ばふばふ」と変形し、移動方向によってポッドも微妙に位置を変える。なにか操作するとそれが絵になって現れる。描き込みは凄いけど、どこか平面的なゲームというのもあるが、動きが自然と奥行きが感じられるものだ。

* * *

セリーグの野球と大リーグを比べるようなものだが、やはりアーケードゲームメーカーは凄いと感ぜさせられる。

イメージファイトは奇をてらわない正統派のゲームだ。ビジュアルや色モノに頼ったり、X68000のハードにおんぶした機能誇示もない。海外モノのような最新のゲームコンセプトも持っていない。それでいてやはり面白い。ここらあたりに厳然とした「力」の差を感じる。

イメージファイトはかなりよい移植がされている。X68000への移植ものでは、ある程度「本物そっくり」は当たり前だ。しかし、たまに、どこがどうというわけではなく、プレイしてみて、どこか魂が入っていないと感じられる移植作品もある。そういったものはちょっと見に似ていても人気が出ない。こと、X68000に関しては移植作品にこういった形而上的な要求がされている。完璧に同じではないにせよ、これに忠えているというのはとんでもないことなのかもしれない。



補習ステージの光景

総評

とりあえずほとんど不満はない。内容はいわゆるパターンゲームだが、パターンゲームにはパターンゲームの美学がある。問題はコンティニューの方法だ。というのも、最近のX68000のゲームはみんな敵が硬い。わざわざ難易度を無茶苦茶に上げているようなものも少なからずある。しかし、最近の若手X68000ユーザーはそういった超ムズのゲームをあっさりクリアしてしまう。で、ソフトハウスはもっと難しいゲームを作り出す。これがまた難しいゲームは敵が硬いゲームだと思いついてるふしがある。困

ったもんだ。アンケートハガキを見ると16, 17歳近辺に新人類がいるらしい。諸悪の根源である。

で、だ。一般人としては、今後X68000のアクションゲームでは難易度設定の際に生年月日を入力させて「15歳から20歳まではコンティニューなし」とか柔軟に調整してほしい、と、最近結構本気で思っている。

10段階評価

グラフィック	★★★★★★★★
音楽	★★★★★★
アクション性	★★★★★★★★
攻略しがい	★★★★★★★★

ひと味違ったシューティングを楽しみたい方に



Yamada Junji

山田 純二

またまたアーケードからの移植作品が登場。このゲームは自機の攻撃力がアップするのではなく、パワーボールというアイテムを使って攻撃していくというもの。もちろん2人プレイも可能、MIDIにも対応している。

ついに発売となったこのゲーム、1987年にテクモから登場したアーケードゲームというのはいま周知の事実だろうけど、X68000版はシステムサコムが移植、開発し、めでたくここに発売となったのだ。

ゲーム自体は縦スクロールのシューティングゲームで、ちと難易度は高め。だからしてももちろん2人プレイもできるようになっている（友達となかよくやろうね）。

アーケード版でもそうだったんだけど、デモ画面を見てまずおやつ？ と思うのは自機のおしりの部分。金魚のフンのようになにやら丸いものがいくつか連なっている。いい、いったいこれはっ!? と大袈裟に驚く人はいないだろうけど、この金魚のフンこそこのゲーム最大(?)の特色なのだ。

金魚のフン、かくしてその正体は！ じゃーん、それはこのゲームにおけるパワーアップアイテム、ガンボールの連なっている図なのでした。ここまで引っ張れば大方予想はついていただろうけどね……。

ガンボールこそ我が命

さて、このガンボールとやら、パワーアップアイテムということでご多分にもれず



X68000用 5"2HD版2枚組 8,800円(税別)
システムサコム ☎03(635)7609

当然いろいろな種類がある（といっても武器関係は5種類なんだけど）。それぞれどういう場面で使用するといちばん有効かがだいたい決まっている。まあ、どれがどんなだっというのにはマニュアルにも書いてあることだし、まず自分でいろいろ使ってみることがいちばんなんだけど。

で、このガンボールを取る手段というのも一風変わっている。特定の敵を倒すと現れることもあるが、主にフリンガーという緑色のサソリに羽が生えたような敵を倒すか、敵から奪い取るにより得られるのだ。フリンガーの持っているガンボールは、フリンガーを撃つたびに種類が変わっていくので、欲しいガンボールに変えてから奪い取ることもできる。だから自分が欲しいと思うガンボールが出るまでガンガン撃ちまくるに限る。取ったガンボールは、自機の後ろにくっついていき、最高15個までつながる。すごいぞー、15個後ろにでろでろしてて、なおかつ敵さんがわちわちやいたりすると。まるで乱視の検査みたい。

しかし、逆にフリンガーにガンボールを奪い取られることもあるにはある。奪い取ったガンボールをまた奪い取られたり、ガンボールを持っていないフリンガー野郎が、ガンボールのみを奪い取ってそそくさと逃げていってしまったりするのだ。うーん、おじさん一本取られたね、などと不覚にも感心してしまった。

さてさて、やっと使用上の注意までごつけたぞ。ガンボールは取った順番につながって、そのとおりに使用されていく。「このシーンではあれを使いたかったのに」などといってもあとの祭り。システムなんだからしゃーないやね。要は、先を見越してガンボールを取る順番、プラス使用のタイミングを考えていけ

ばいいのだ。とはいえキヨタ君じゃないんだから先のことなんてわかんないけどさ。結局何度もトライしたもん勝ちなのだ。

ほかにはなにがある？

さっきも書いたガンボール奪い合いシステムだが、応用編として2人プレイのときにも互いのガンボールの奪い合いができるのが面白い。2人でやれば攻撃力倍増で、戦いは比較的楽になるけど、2人して動き回っているといつの間にかガンボールが相手に取られていたりして、せっかく攻撃しようにもガンボールがない、なんてことも起こる。しかし、この性質をうまく使えば、逆に戦いを有利に進めることもできるのだ。

ガンボールが尽きてしまったときには、相手から分けてもらうこともできる。が、いちばんの使い道はガンボールの組み替えだ。ガンボールは取った順番につながるから、使いたいガンボールが後ろにある場合は、それまでのガンボールを無駄に使うか、それまで我慢しなくてはならない。そんなときには相手にガンボールをすべて渡して、そのあと自分の使いたいガンボールを相手からもらえばいいのである。

グラフィックについては蟲、蟲、蟲の嵐だが、それほどグロいと感じなかった。割と明るめな色合いで、ドロドロ感はない。



ずらずらと金魚のフンのようにガンボールをつなげちゃえ！

逆にライトすぎると感じたほどだ。ゲーム中で、僕がいちばんグロイと感じたのは、緑色の原色卵を撃つと、でろりん、と中身が飛び出てくるヤツ。あれはグロいよね。

音楽は、なんだか妙。シューティングゲームの音楽は、だいたいにおいてカッコよく、ノリノリのものが多いが、このゲームのBGMは静かめの曲が使われている。特に1面の曲なんかはのどかすぎるほどで、ちょっと気が抜けてしまった。でも、徐々に面が進むにつれて重々しい感じになっていき、はじめのうち感じていた違和感はなくなっていったけど。このゲームはMIDI（MT-32系の音源）にも対応しているの、持っている人はこの妙に雰囲気のある音楽を存分に楽しんでもらいたい。

さて、と、それでは、それぞれの面を見ていくことにしようか。

ステージ解説

1面：のどかな音楽にのって、このゲームのコツをつかむにはちょうどいい面である。敵の攻撃もさほど激しくないの、取れるガンボールは残さず取り、あとあとの面に備えたい。この面のボスは、でっかい襟巻きアザラシ君。攻撃も単調だし、さっさとかたづけてしまおう。あまりもたついていると、華麗な飛び込み技でやられてしまうから、注意すべし。

2面：森の上をがしがし進んでいく。1面で貯めておいたガンボールをうまく使って敵を倒していこう。面の半ばにくるといきなり高速スクロールが始まるが、地形が速くなるだけなので、あわてず冷静に。この面のボスは、穴から飛び出たひとつ目親父2人。目玉を狙って弾を打ち込むべし。一定時間おきに、ぶっといレーザを撃ってくるが狙いは甘い。ひよいひよいと死角で待機していればらくちん。

3面：今度は洞窟の中を進んでいくことになる。狭い洞窟内であるから、アイテムによってはそれほどの効果を得られないものが出てくる。障害物で、攻撃力が半減さ



アザラシ君のダイビング、よっけるー！



このドロドロした暗さがたまらない

れてしまうのだ。3面にはボスキャラはいないが、最後のほうで左右から現れる鳶には注意が必要。撃っても撃っても際限なしに出てくるので、ここは真ん中あたりのいちばん上（変な表現！）に逃げてしまおう。

4面：洞窟を抜けるとそこは嵐であった。派手な雷鳴とともに画面がフラッシュし、なかなかニクい演出である。個人的にはいちばん好きな面だ。敵はうじゃうじゃ、弾もうじゃうじゃ吐いてくるため、もう大変。障害物がないだけまだましだが、このあたりからシューティングのうまい人とそうでない人の差が表れてくるだろう。ボスはぶよぶよした塊。ときどき現れてくる発射口を狙って撃て！

5面：嵐を抜けるとそこは空であった（もういいって）。ここでいちばんやなヤツは雲のなかから突然現れるコウモリおじさん。とんでもねー、よけれねーぞ、と文句いってもしようがない。ただただ、よけろーっ。と、ひたすら敵の猛攻をくぐり抜けると浮遊戦艦の登場だ。これの弱点は、左右に見え隠れする砲台である。ここでは、左右どちらかを集中的に攻撃するのではなく、交互に攻撃したほうがいいだろう。

6面：戦艦をやつつけるとそこはジャングルであった（しつこいなあ）。やっとの思いで5面を抜けたと思ったら、さらにキツイ面が僕を待っていた。ハンパじゃない攻撃に人食いチューリップ。もう、勘弁してくれの世界である。根性のドット単位のすり抜け技を全開させてクリアを目指そう。ボスは百本足のムカデちゃんだ。

……以下7面8面と続く……だろう。がんばってください（それしかいえない）。

緊張感と面白さの度合い

このジェミニウイングというゲーム、アーケードからの移植とあって、バランスも取れているし、シューティングゲームマニアにとって、ゴキゲンなソフトであるといえる。高い難易度、独特のグラフィック、加えてパワーアップのシステム、どれを取っ



これがウワサの人食いチューリップだ

ても八分どおり満足できるものだ。

多くのパワーアップシューティングゲームでは、アイテムを取ることに自機がパワーアップされていく方法をとっている。このようなタイプのゲームでは、途中で死んでしまうとそこからの復活に超人的な技術が必要とされ、プレイヤーは、死んでしまったのはもともともない、という緊張感にゲームを楽しむことになる。

しかし、このゲームはアイテムの攻撃力によりパワーアップしていくのである。さらにアイテムを使うタイミングと取る順番、この2つがよりプレイヤーに刺激を与えているのである。もちろん、前者のような緊張感もあるが、その緊張感にアイテムの使用タイミングという別の要素により、プレイヤーにゲームを面白いと感じさせているといえるだろう。

さて、マニアではない普通の人にはどうであるかということ……、ちょっと難しいかもしれない。最初のうちはともかく、後半戦が異様にキツくなっているから、途中で投げ出してしまう危険性が大。まあ、それなりに楽しめるかもしれないが。

マニュアルにも遅れた言い訳がたくさん載っているけど、そうして時間をかけたぶん十分な出来栄となっているので、プレイする側としてはとりあえず安心していい。といったところでサコムさん、次のアトミック・ロボキッドに期待してますよ。

ヘイヘイ！ ジェミニ！（総評）

サコムさん、コンティニュー3回なんてきつすぎます。せめて、8回ぐらい許してください。文句はそれぐらいで、アーケードゲーム版との違いは、僕の見た限りないといっていいいでしょう。よい出来です。この調子で次回作もがんばってやってもらおう。いけいけ、GO、GO！ サコムさん！

（5段階評価）

難易度	★★★★
グラフィック	★★★★
ガンボール	★★★★
移植度	★★★★★

機能満載! 横スクロール



Kageyama Hiroaki
影山 裕昭

デビュー作というのは、作るほうにも買うほうにも「期待と不安」の入り混じった気持ちを抱かせるものですね。たいていの場合は「不安」のほうが多くなりがちだけど、このゲームは期待してもいいと思いますよ。

新潟のソフトハウス「エグザクト」、といっても知らない人が多いでしょう。それもそのはず、このソフトがデビュー作なんですから。それでも、すでに開発途中の画面写真があちこちの雑誌で紹介されているので、ゲームタイトルくらいはチェックしていた人もいると思います。ちなみにNAIOUSはナイアスと読みます。

どんなゲームなのか

このソフトは基本的には横スクロールのシューティングゲームなのですが、縦スクロールがおりまざっているステージもあります。このときは機首が画面の上を向くのではなく、機首は右を向いたまま垂直上昇（あるいは降下）していきます。

ステージは全7面で、各ステージの最後にはボスがしっかりと待ち構えています。このテのゲームにはもうなくてはならないコンフィギュレーションももちろんあります。EASY, NORMAL, HARDと3段階の難易度調整ができ、難易度を上げるとアイテムが取りにくい場所に移ったり、敵の出現パターンがいやらしくなってきます。また、ソフト側で連射するように設定でき

るので、少なくとも片方の手は疲れません。ステージ3まで面セレクトが可能、自機は最高5まで増やせ、コンティニューも7回までできるという親切さもいいでしょう。

兵器は最初はバルカンのみですが、アイテムを取ることによって、ワイド、レーザー、ホーミングなどが、Aボタンで選択することで使うことができるようになります。それぞれの兵器は2段階までパワーアップします。アイテムはそのほかに、1UPアイテム、シールド、選択している兵器のレベルを最高にするものなどもあります。

また、これとは別に各ステージで味方と思われる未確認飛行物体がオプションを投下してくれます。これにはそれぞれ特長のある4種類があり、攻撃範囲をぐんと広げてくれます。しかし、自機が破壊されると選択していた兵器とともに、オプションもなくなってしまいます。

ここまで読んで「ああ、ありがちなゲームだなあ」と思った方も多いでしょう。しかし、見た目だけで判断するのはよくありません。このゲームの長所はシューティングゲームになくてはならない「スリリングなゲーム展開」が徹底されているということです。画面全体に細心の注意を払ってプレイしている自分にふと気づくほどです。実はこれが簡単そうでいちばん難しいこと。

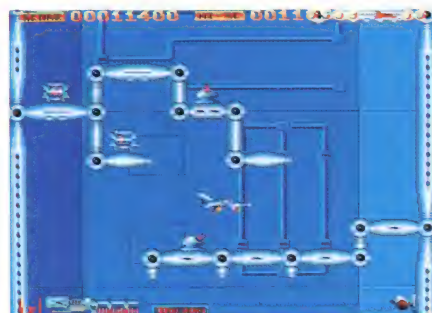
1ステージを通して敵キャラの出るタイミングが実によく考えられていて、これが効を奏しているようです。太いレーザーで攻撃してくる敵、ワープしてくる敵、突然後ろから飛んでくるミサイル。どれもこれもプレイヤーの度肝を抜いてくれます。

これこそがシューティングの基本であり、ゲームデザインのセンスが問われるところでしょう。この基本が崩れているシューティン

グはほかの部分がいくらすばらしくても、意外にあっさり飽きてしまうものです。その点ナイアスのゲームデザインは誉めてもいいでしょう。

ところで、パズルゲームとシューティングゲームが苦手だという（で）君にEASYで遊んでもらったら、ファーストプレイでステージ1をクリアしました。面セレクトできるステージ3までは難易度も低くしてあるように感じます。ただし、ステージが進んでいくと「もうお手上げ」というくらいの難易度にもなります。

業務用のシューティングゲームをかなり意識したソフトなので、ゲーセンに足を運ぶ人はどんなゲームを参考にしているかすぐにわかることと思います。そのためか、ゲーセンにあまり行かない人のほうが夢になる度合いが強いです。



ここは縦スクロール。アイテムを取れ!



二重ラスタスクロールのステージ3。オプションだ!



X68000用 5"2HD版2枚組 8,700円(税別)
エグザクト ☎025(247)9160

なにが起こったのか

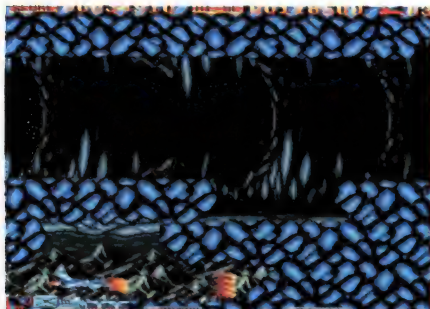
ナイアスは自機の名称ではありません(自機はASPというそう)。ナイアスは地球環境の管理、さらには太陽系全体の環境管理を行う推論型コンピュータの名称です。ナイアスの始動直後、地球環境は改善に向かいましたが、その数年後には、人類の活動がナイアスの浄化作用を上回ってしまったのです。そして西暦2143年、あまりにも愚かな人類に対してナイアスはひとつの答えを導き出したのです……。

このゲームの目的はズバリ、ナイアスを破壊することにあります。たとえゲームのなかであっても、地球環境の話が絡んでたりして考えさせられるものがあります。

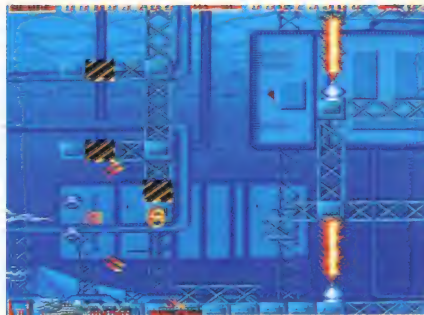
第1任務は「木星のコロニーを奪還すること」というわけで、宇宙空間からゲームが始まります。最初はほとんど雑魚キャラ。しばらくすると上から巨大な戦艦が……、「えっ、もうボスキャラ?」と思ったら大間違い、これがコロニーです。敵キャラ相手に戦闘を繰り広げている間も、コロニーが上下にフワフワと動いていて、なかなかニクい演出をしてくれます。

スクロールが止まるとコロニーのゲートが開いて、今度は縦スクロールに変わります。ジェット噴射も下方向に変わり、自機が上昇していることが容易に理解できます。そして、コロニー本体に入るとまたまた横スクロール。背景もコロニーの感じがよく出ていて、ガンダムのサイド6を思い出してしまいました。ステージ後半で自機の周りにワープしてくる敵には注意。それを抜けるとボスがやってきます。まあ、楽勝でしょう。

ステージ2は「ASTER」のエンジンを破壊して、冥王星への激突を未然に防ぐことが目的です。途中に出てくるハンバークみたいな中ボスは安全地帯があるから、それさえわかれば恐れることもありません。最後のボスもそんなに難しくはないでしょう。ステージクリアすると背景がラ



アイテムに気をとられるとこんなハメに……



ステージ5は海の中。背景がゆ～らゆ～ら

スタスクロール。思わず唸る演出だけど、この時点で僕はまだこれがなにを表現しているかはわかっていませんでした。

目をひくラスタスクロール

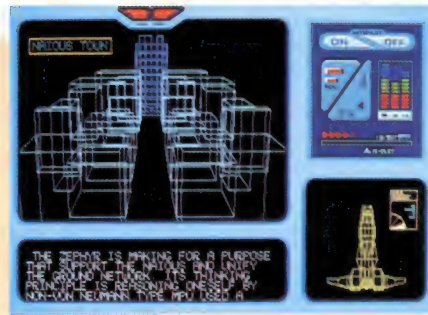
ステージ2は我々の目先を変えるための罠だったのです。「ASTER」の爆発によって作り出されたねじれた空間にASPは吸い込まれてしまいます。あのラスタスクロールは空間のねじれを表現していたのです。そして、この空間を脱出することがステージ3の目的です。

このステージでは2枚のBGが別々にラスタスクロールしていて、とにかくすごい。ハデハデな背景と、ラスタスクロールのおかげで敵の見にくいことといたら……。おまけにそれを保護色として巧みに利用している敵もいます。このステージでは、後ろから突然敵が出現するのでかなりビビります。そしてボスが結構手強い。ま、これ以上ネタをばらすとつまらなくなるからやめときましょう。

だんだんムズク

ステージ4あたりから徐々に敵の攻撃が激しくなってきます。ここは洞窟であちこちで落盤が発生するので、あんまり前のほうには出ないほうがいいと思いますよ。

書きたいことは山ほどあるんだけど、あんまりばらすとこれから遊ぶ人がつまらない思いをするだろうし、難しいところだな。



ステージごとにミッションが解説される

とりあえず中ボスがいます。これがまたすごい。いきなりラスタスクロールして登場するんだもん。そしてレーザーとホーミングを好き放題撃ちまくると、半透明になってラスタスクロール。ワープしてるっ!!

ワープ中はヒットチェックがない。そしてラスタスクロールが止まると、半透明も終わってぐるぐる回りながら攻撃。初めてこれを見て平常心でいられる人はいるのかな。いやいやすばらしい演出です。

ステージ5は海底シーン。今度は海底の建造物が崩れ落ちてくるので、やはり前に出るのは危険です。ここでも海底の様子をラスタスクロールで表現していて、これがなかなかいい味を出しています。しかし攻撃が前にもまして激しくなっている。ジョイパッドしかない僕は右に動こうとして右下に動いてしまったりして、操作もままならない。こりゃ、ジョイスティック買ったほうがいいかな。

これぐらいにしておこう

シューティングゲームの原稿は必要以上に内容をばらしてしまうと、ゲームをつまらなくしてしまうのですごく書きにくい。ステージ6のボスの動きがすばらしいので、ぜひとも見てもらいたいのですが、いったい、そこまでたどり着ける人がどのくらいいることや。とにかくエグザクトさんのデビュー作は立派。これからが期待できるソフトハウスでしょう。

ゲームバランスが命

ナイアスは全体的によくできているゲームだと思います。ただ、既存のゲームを参考にしている点が多いので、ケチをつければ、「オリジナリティがまるでない」ということはいえます。それを抜きにしても、ジェミニウイング、イメージファイトといった大作シューティングがほぼ同時期に発売されるので、ちょっとかわいそうな気がします……。

背景のグラフィックはとてもきれいで、これはかなりよろしい。もうちょっと自機をカッコよくしてくれたらもっとよかったんですけどね。なにはともあれ、ナイアスはプレイしていて楽

しいゲームなので売れても不思議ではありません。だからこそ、売れたときにはそれで満足せずに「ソフトハウスとしての真価が問われるのは、2本目の出来いかんだ」ということを忘れないようにがんばっていただきたい。なかなかいいセンいってと思いますから。

7段階評価

キャラクタデザイン	★★★★☆
ゲームバランス	★★★★★
サウンド	★★★★☆
ラスタスクロール	★★★★★
お買い得感	★★★★☆

THE SOFTOUCH SPECIAL

敵はどこだ! ローラーダッシュ!

●FZ戦記 アクシス



Saitou Susumu
斎藤 晋

“巨大で重い”戦車はすでに必要でなくなった。そして、NAP(New Age Powered suits)が誕生した。ハワード・ボウイ、NAPを操り、戦略上重要なポイント、ブラッディ・アクシスを占拠、破壊せよ。それが指令だ……。

ローラーダッシュをかけると荒れた街並がみるみる画面の外へ流れていく。視界に目標の姿はない。ターゲットキャラクターは残り5機。出てこないなら誘い出すまでだ。が、物陰から雑魚がオレンジの弾を撃ってくる。邪魔なんだよ! スティックを入れ替えTWINを連射する。

NAPと呼ばれる機動兵器がお馴染みウルフサウンド(MT-32にも対応)に乗ってスピード感、重量感溢れる戦闘アクションを展開する。それがFZ戦記アクシスだ。

これがNAPの威力だ

アクシスの見どころのひとつ、それは画面構成にある。すなわちクォータービューというやつだ。視点を斜めにするとなにが面白いという立体感が出る。昔のZAXONなんかも当時は3Dものとして扱われていたように思う。透視図じゃないから遠近感こそないが、クォータービューというのは3次元空間を表現するためのモデル化のひとつだったわけだ。

ゲーム全体は8つの面に分かれ、それぞれが2面構成の合計16ステージだ。それぞれ一定のパターンで現れる将校キャラクタ

ーを規定数倒せばステージクリアとなる。

NAPはふつうに操作するときはトコトコと歩くのだが、スティックを同じ方向に2度倒すと、高速移動モードに入る。これがローラーダッシュだ。画面を見てもええかわかるとおりNAPのキャラクターは画面中央に表示されている。これが高速で移動するといったらどんな感じが想像してほしい。実際に動くのはキャラクターじゃなくて画面のほうである。それもびゅんびゅんの8方向スクロールだ。ローラーダッシュの状態ではスティックをコントロールすればちょっとした「めまい」を感じることができる。これがなんといってもこのゲームの魅力だな。また、敵に接近するとNAPは自動的にパンチを繰り出す。すごいぞ! これが強力なのだ。

最初のステージは市街戦だ。目前にタンクが3台迫っている。が、こいつの弾には当たる心配がない。やつの砲台は回らないのだ。ここは背後からローラーダッシュで接近し必殺のパンチを繰り出そう。続いてまた3台。楽勝だ。次は戦闘ヘリがくる。建物の上を飛んでいる間は攻撃ができないから、道路の上空に降りてくるときを狙い撃ちする。でもヘリに気を取られていると思わぬ雑魚キャラの攻撃を食らうはめに。操作法に慣れるまではちょっと苦戦かな。

使える武器は右手のハンドウェポン(Bボタン)と背中のオプションウェポン(Aボタン)があって、最初は各々にNORMALと呼ばれるユニットがセットされている。実はユニットは汎用性あって取り付け場所によって働きが違う。ハンドウェポンとして用いれば弾の威力はさほどでもないが弾数に制限がない。逆にオプションウェポンに装着すると効果は大きいが使える回数に制限がある。

たとえば、BAZOOKAというユニットはハンドウェポンとして使うと、前方のみの連射だが、オプションウェポンとして使うと扇型9方向弾になるといった具合だ。また、武器として使わないときは各部にシ

ールドとして装着される。つまりユニットの残りが耐久度となるってわけだ。

ユニットは標準装備のほかに戦闘のなかで手にはいるものも多い。特にオプションウェポンで使うと楽しいのがいっぱい。炎が円状に広がるナパームや誘導弾、残酷なワイヤー攻撃、なぜか爆撃機の援護がくるものまである。しかも、マニュアルにはまだ手に入れたことのないものが載っているではないか。弾数に制限はあるが、せこい考えは無用。ユニットは被弾するたびに失ってしまうのだ。弾切れとなったユニットもシールドとしては有効なのだからどうせなら積極的に使っちゃおう。

さて、多くの敵は弧を描くように攻めてくる。動きが読めれば相手の慣性系に入り込んで簡単に倒せるが、ほとんど平行移動のような関係になるとイライラする。違う方向に撃ちたいときは一瞬ボタンを離してからスティックを入れ直すわけだが、肩に力が入っているとうまくいかない。広いエリアでは向きを変えずにローラーダッシュ



1面のタンク。かわいいけどちょっとマヌケ



強制スクロール面。巨大な装甲車が4台来る



X68000用 5'2HD版3枚組 8,800円(税別)
ウルフ・チーム ☎03(5273)4795

をこまめに利用して敵の動きに追従できるようになると楽しい。

ま、そんなこんなで1面の戦闘ステージをクリア。最後のヘリを撃墜したら周囲はたちまち誘爆の嵐となっておしまいだ。

1面後半のステージは、斜め方向の強制スクロール面。巨大な装甲車と対決だ。攻撃してくる砲台野郎をやっつければいい。2面は原野。3面は神殿、4面は洞窟と続く。



目の前で弾を炸裂する4面のボス

エレベータの面。スピード感、浮遊感に満ちた楽しい面だ



やっぱり拳が頼りだね

僕が一番気に入ったのは5面前半。高速で上昇するエレベータのステージだ。周囲を取り囲むようにヘリが飛来し、前後のエレベータが徐々に速度を合わせてくる。もちろん敵キャラが乗ったやつだ。ここまでこれようになれば、NAPの操作もそれほど不自由はない。リングの誘導弾も2面のボスキャラでかわし方は修得済みだ。

後半は文字どおり間抜けな大砲が待っている。あまりの間抜けさに哀れを感じてしまった。哀れな大砲よ、安らかに眠れ。

6面は1面や4面と逆で前半が強制スクロール面。敵は1匹。しかし、巨大な弾丸を一気に吐き出すように撃ってくる。冗談

ではない！ と思ったが、これが不思議とよけられるのだ。まるで自分がうまくなったかのような気分に酔いながら、撃破。

後半には巨大な建造物のまわりを囲む2重の動く歩道が。こいつの上ではベルトの向きに逆らって歩いてもほとんど進まない。じたばたしていると思わぬ被弾にあう。

7面。撃破しなければならない将校キャラは25と最も多い。この面では結構新しいユニットが手に入る。とはいえ、ゲームはまもなく佳境であるから、このあとの敵に備えてユニット（シールド）を無駄に失わないようにしたい。後半はどうやら要塞最後の中核のようなのだ。まわりをガードする敵の動きがかなり素早くなっている。

さあ、敵要塞内部をことごとく破壊した

ボウイの前に最後の敵が立ち塞がる。敵はたった1機の赤い機動兵器だ。しかし、速い！ NAPと同じ運動能力を持っているようだ。回り込もうとしてもなかなか後ろを取らせてはくれない、どころか、うかつに近づくと逆に後方に回り込まれてしまう。反射的にローラダッシュで後退だ。どうする？ RAYをオプションウェポンに装着。

8方向レーザーで死角からの侵入に備える。これなら敵は

よけられない。が、そうした思惑もどうやら期待倒れに終わった。敵はちょっとやそっと被弾してもびくともしない。こちらの火力じゃ倒せないじゃ？ そう思ったとき、しばらく使っていなかった戦法を思い出した。仕掛ける前には一発も食らいたくない。コーナーの狭い通路に赤い影を見つけるやいなや、私はスティックを斜めに入れ、ローラダッシュで突進していった。

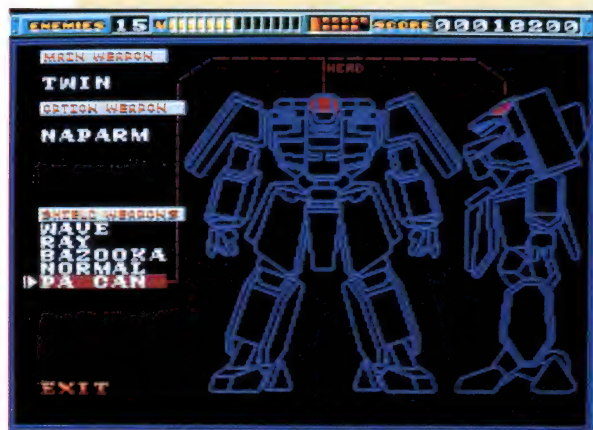
お陰さまでエンディング

アクシスは、ステージのバリエーションが楽しめる。強制斜めスクロールの面やエレベータの面はやっていて本当に楽しい。が、市街地や広いフィールド面ではパソコン用ゲームにありがちな冗長さが目立つ。人によっては最初の市街地でほとんどターゲットキャラが出てこない状況に直面する。僕より先にプレイしていた何人かは敵の居場所がわからないと途方に暮れながら街中をさまよっていた。雑魚キャラを30匹ぐらい倒せば出てくるんじゃない？ なんて憶測もあったほどだ。

実は、市街地をなぞるように一定方向に探しても敵は出てこない。若干コツがあるのだが、ちょっと挑発的な動きが必要なのだ。逆に敵の誘い出し方さえわかればゲームは俄然テキパキと進行する。

あと、残念なのは、オプションウェポンが存分に生かせていないことだ。僕の場合、3面で拾えるPA CANをハンドウェポンで使えば（これがかなり強力）あとは特にこれが必要といったものを感じない。ユニークな武器（使って楽しいもの）が結構あるだけにちょっともったいない。

というわけで、EASYモードならアクションのかなり苦手な人でもクリアできるはずだ。ラストステージは「脱出」。ひたすらよけるだけだが、そのままエンディングへとつながるあたりの演出もいい。よし、今度はHARDで挑戦しようかな。



ユニットセレクトモード。カッコいい

この路線でガンバッて！

X68000にはアクションゲームがたくさんあるが、評価の高いのはアーケードからの移植ものが多い。逆に、X68000で開発されたオリジナルゲームでアーケードに移植しても通用するものといえはちょっとお寒い状況だ。前作グラナダは少数ないさういうゲームだったように思う。このアクシスも基本的なゲームデザインの面ではアーケード指向のゲームだろう。クォータービューとか8方向スクロールというのは必ずしも万人受けしないかもしれないが、ウルフ・

チームさんにはこういうとんがった部分でこれからも意欲的な作品に挑戦してもらいたい。

もちろん、グラナダ、アクシスとくれば、こっちだってそれ相応の期待をしなくちゃね。次回作もよろしく。10段階評価

グラフィック	★★★★★★★
サウンド	★★★★★★★
スピード感	★★★★★★★
イライラ感	★★★★★
オープニングデモ	★★★★★

真の戦場の姿とは？



Yamada Junji
山田 純二

いつもちょっと変わったゲームを発表してくれるアートディンク。今回のこのゲームは戦場をよりリアルに表現するために、いろいろな新しい手法が取り入れられている。さて、その結果やいかに？

「機甲師団」は戦闘画面からHEX画面を排除し、さらにリアルタイムで戦闘が進行していくシミュレーションゲームである。舞台は第2次世界大戦におけるヨーロッパ戦線の地上戦。プレイヤーの最終目標はノルマンディから上陸した連合軍第7機甲師団7大隊を率いて、全12の戦場で待ち受けるドイツ機甲師団を撃ち破り、ヨーロッパ戦線を制圧することとなっている。

シミュレーションというのは、ある事象をモデル化し、そのモデルを操作することによって現実の事象を理解し予測を深める技法である。この場合の現実事象というのは地上戦であり、ゲームの性質上、モデル化するにあたってはできるだけ、より現実に近いものにしておくべき。もちろん、すべての要素を盛り込むのは不可能なので（仮にできたとしても、それがゲームとして成り立つかは疑問）、たとえば光栄に代表される一連のシミュレーションゲームでは、登場するキャラクターを現実に近づけるという方法を取るなどしている。

それに対してこのゲームの場合には、HEXによって画面を細切れにすることを廃止した戦闘方式とリアルタイムなゲーム

進行を採用することで、プレイヤーにより実戦に近い雰囲気を感じさせようとしている。

はずかしながら僕はシミュレーションゲームというもの、これまで一度もプレイしたことがない。どちらかといえば、短時間で終わるシューティングゲーム専門であった。1プレイに数時間、もしくは十数時間もかかるようなゲームは敬遠していたのである。今回、なぜこのようなゲームを選んで（自分で買った）しまったのかというリアルタイムシミュレーションという、売り文句を聞いたとき、このゲームなら純粋に戦闘を楽しむことができる（あまり時間もかからずにね）のではないかと、思ったからである。

そして、最終目的を達成した現在、その思い込みは果たして正しかったかどうか……は最後に述べるとして、まずはこのゲームの内容を見ていくことにしよう。

基本構成

まず、取り込み画像をふんだんに使ったオープニングが始まる。そして、それが終わると作戦内容が表示され、画面は師団司令部（GHQ）に移る。これは部隊全体の作戦行動を指揮するための画面である。

一番最初に行くことは、ユーザーディスクの作成。マニュアルどおりに作成すれば、いよいよゲーム開始である。そうそうマニュアルは60ページと比較的短いので、ゲーム開始前にしっかりと読んでおくことを勧める。

最初は、爆発マークのついているマップAに部隊が上陸しているので、とりあえず戦場に移動。すると、うじゃうじゃと細かいキャラクターがわんさかいる。これがプレイヤーの操る連合軍第7機甲師団7大隊であり、各キャラクターがどのユニットであるかは、図1にまとめて書いてあるので参考にしてもらいたい（なぜかマニュアルには載っていないのでご参考に）。

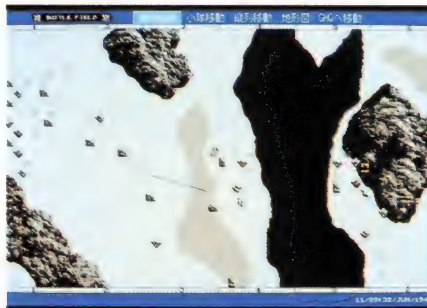
マップ全体を把握したい場合には「地形

図」というメニューバーをクリックするとマップ全体の縮小マップと四角い枠が表示される。この四角い枠が現在表示している範囲で、移動させたい場合にはこの枠をずりずり引きずっていくか、あるいはメイン画面上に表示された格子線をドラッグして（Z'sSTAFFのファイルメニューのように）ずりずりと引きずっていけばよい。また、これらを実行している間は時間が止まっているので、心置きなくマップを眺めることができる。

さて、今度はユニットの移動。これは、移動させたいユニット、または小隊を左クリックし、移動させたいポイントでもう一度クリックすればユニットと目的地が線（移動ルート）で結ばれる。第4の目的地まで指定でき、右クリックすることで確定される。進行中に障害物、あるいはほかのユニットがあるとその場で止まってしまうので注意。また、ユニット自体を右クリックすると、部隊情報を見たりそのユニットに作戦行動を与えることができる。その結果、敵にぶつかるると自動的に戦闘開始となる。マップにいる敵部隊を全滅させると（少し間の抜けた音楽が流れて）、そのマップを制圧したこととなる。

補給こそ命

戦闘に勝利したからといってはと一息つくひまはない。今度は補給である。弾薬や燃料の残存量も気にしなくてはならないが、問題は兵員である。中隊司令部から小



部隊の移動先をベクトルで指定



X68000用 5" 2HD版3枚組 9,500円(税別)
アートディンク ☎0474(77)7541

隊の補給は3時間ごとに自動的に行われるが、大隊司令部から中隊司令部の補給は作戦行動によって行わなくてはならないのだ。仮に兵員、物資要請したとしても届くまでには約2時間もかかってしまう。部隊を動かして、実際に戦闘が行われるまでの待ち時間をうまく使って各部隊のチェックを行い、必要ならば補給の要請をしておかなくてはならないのである。このことをうっかり忘れてしまうと意味のない待ち時間が発生してしまう。リアルタイムにゲームが進行していることを決して忘れてはいけない。

補給に関係してもうひとつ考慮しておかなくてはならない点がある。それは、各部隊にいる士官で、部隊全体の戦力は将兵のいる残存部隊数によって決まるのである。部隊にいる士官が死ぬと部隊が全滅してしまい、戦力が低下してしまうのだ。だから、将兵が死んでしまったらGHQ画面の部隊情報で新しい将兵を任命して、部隊を再構成してやらなくてはならない。残存将兵の数は限られており、あとからは新しく追加できないので、部隊が危なくなってきたら前線から撤退させるか、降伏させたほうがいだろう。

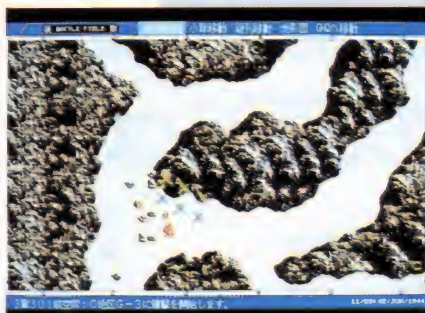
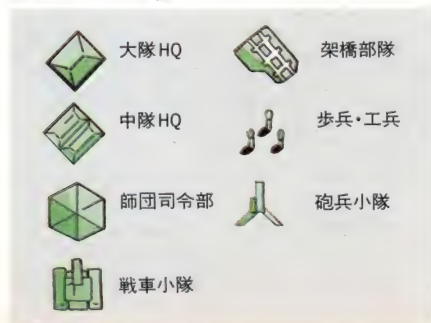
作戦行動は「砲撃要請」などが中隊司令部か大隊司令部から行うことができるが、小隊は「降伏」しかできないことも覚えておかなくてはならない。

航空部隊を使え

第7機甲師団以外に作戦行動を要請できる部隊として、第3軍所属の航空部隊がある。この部隊は完全に独立しているので、いくら呼び出しても第7機甲師団には影響がないから、非常においしい部隊である。

要請の方法はGHQ画面でメニューバーのスケジュールをクリックし、要請内容、場所、時刻を設定すれば律儀にスケジュールどおりの行動してくれる。しかし、一度にスケジュールを設定すると司令部のほう勝手に時間調整をしてしまう。一度決めたスケジュールは取り消せないの、役に

図1 ユニット表



航空部隊からの爆撃

立たないように思える。しかし、実は抜け道がある。スケジュールをためてしまうと時間調整されてしまうのなら、ためなければよいということ。それだけである。

たとえば、マップBに爆撃要請をしたい、ついでに兵員の補充もしたいという場合には、とりあえず最短時間に爆撃要請を行うようにスケジュールを設定し、戦闘場面に移ってから爆撃されるまで待つ。爆撃が終わったらすかさずGHQ画面に戻り、今度はスケジュールに兵員補給要請を設定する。この場合も時刻は最短時間になるようにする。すると、10分後に設定できる。

操作は面倒だがこれは便利。物資補給は、わざわざ中隊からの補給を待つより、航空部隊からの物資空輸を使えば、一発でマップにいる部隊に補給が完了してしまうので、これまた便利である。

28時間の死闘

以上のことを基本に、ガシガシ戦闘を進めていけば、必ず、勝利を手にすることができる(本当かなあ)。結局、制圧までの総プレイ時間は28時間ぐらいたった(個人差はあるだろうけど)。じっくりと腰を据えればもっと楽しめるだろうが、僕の場合時間的な制約があったので結構キツかった。

最初の思い込みの「割と時間がかからないだろう」というのはちょっと違ったようだ。ここまで時間がかかった最大の原因は、ゲームの進行が遅いということ。ゲームはだいたい現実時間の10倍ぐらいで進行して



勝利目前、敵は師団司令部のみ

いく(つまり6秒でゲーム中の時間が1分進む)のであるが、もし、ウェイトをかけてわざとそうしているならゲームスピードを自由に設定できるようにすべきである。

「それでは忙しすぎてゲームにならないよ」と勝手に判断されては困るのである(うーんわがまま)。某A.T.氏曰く「プレイヤーをあわてさせるくらいがちょうどいいんですよ」。まったく賛成だ。

文句ついでにもうひとつ。司令部からのメッセージ表示に少し問題があるんじゃないでしょうか。一番下のラインに1行表示されるだけなんて……。履歴機能を付けろとまではいいませんが、せめてウィンドウを開いてよ。しかも、マウスをクリックしてしまうと消えるし、おいらとっても悲しいぜ。

しかし、まあ戦闘のシステムについては気に入りました。たしかに、無味乾燥なHEX画面よりも実際に戦闘をしているという雰囲気はよく出ている。マップのグラフィックも渋く決まっているし、効果音はサンプリングを使ってこれまたよし。戦闘シーンでの味方の部隊が全滅したときのくやしき、敵の将兵を倒したときの快感。そして、補給を待っているときのいつ攻撃を受けるかというスリル。これはリアルタイムだからこそ味わえるものであろう。

良くも悪くもアートディンクの個性が光るこのソフト。短気な人にはちょっと辛いかもしれないが、時間的に余裕のある人にはお勧めである。

人間辛抱だ

このゲームにはBGMというものが無い。まあ、オープニングミュージックを聞くかぎり、BGMがなくて正解だったかもしれない(ちょっときつかな)。本文でも述べているとおり、プレイ時間はかなりかかる。その間、同じ曲をエンドレスで聞かされてはたまらないかもしれないということである。耳がさびしい人は自分の好きな音楽でもかけて、プレイするのがいいだろう。さすがに、アーシアンCDはいいいちであつた(ああっ趣味がバレル)。それでもまだひまな人

は漫画でも読みながら、もしくはプログラミングでもしながらのんびりやるというのがよい(体験談)。あまり長時間ほったらかしにしておくと、どうなっても知らないけど。

5段階評価

戦闘	★★★★★
グラフィック	★★★★★
効果音	★★★★
アートディンク	★★★★★
総合	★★★★★

ひさびさの3Dシューティング登場



Yoshida Kenji

吉田 賢司

あの「メタルサイト」を作った CROSS-WONDER のスタッフが開発したというオリジナル3Dシューティングゲームが、ビクター音産から緊急発売されました。その技術が今回もうまく生かされているようです。MIDI対応。

異次元空間を中心とした舞台でくり広げられる壮絶な戦いと、MIDI 対応（といってもまた MT-32 だけだね）の高品質な BGM が最大のウリ。毎日掲示板から「休講」の 2 字を探すファンキーな大学生も、冬季講習の申し込みで学校を休む中高生も、日夜漢字書き取りに明け暮れる小学生も、ちょっとと汗ばんだその手を休めて単純明快のシューティングでもどうです？

ヨカッタね、スコアもついたよ

ゲームは、バイオパワードスーツ「ウィネス」を装着した女性自キャラを操作し、前後から迫りくる敵を撃ち落とすというもの。また、4つのオプション兵器のうちひとつを装備でき、これを各面スタート時に選択します。ゲームはエネルギー制を採用、これは面クリア時および一定数以上の敵を破壊すると順次補給され、逆に敵からの攻撃を受けたり前述のオプション兵器の使用によって消耗します。さらに、自機を飛翔させることによって常時消費されます。これは制限時間と考えてもいいでしょう。

自機は飛行スピードを調整でき、単にスクロールスピードを上げ下げするものにと



これがエリア 1、異次元空間だ！

X68000用 5"2HD版2枚組 8,800円(税別)
ビクター音楽産業 ☎03(423)7901

どまらず、なんと後退もできるのです。この機能により、バラついて出現する雑魚キャラや高速飛来する敵などこすりそうなヤツは、通常より長い間画面にとどめておく確実に破壊することができます。

そうそう、西川善司氏が「メタルサイト」のレビューのときに「X68000のオリジナルアクションゲームは、なんでスコアがないんだ」と叫んでいましたが、なんでも彼の意見を尊重して今回は、そういった表示関係を特に大きくした、ということです。善さんヨカッタね、スコアついて。

各ステージをちょこっと紹介

ゲームをスタートすると、一面真っ赤の亜空間。ほとんど練習同様なので、ボス出現までスピードを最高速に保ってバンバン雑魚キャラを落としましょう。このステージでは特殊兵器を装備する必要はないでしょう（そうそう、特殊兵器を装備しなければ、面スタート時に150ポイントのエネルギーボーナスが加算されますよ）。

ステージ2は一変して近代都市へ。このステージもステージ1同様ラクにクリア。

ステージ3から少し難しくなってきます。高速飛来する堅い敵は、スピードをやや後退気味まで落としてから攻撃するとよいですね。ボスキャラが結構堅いので、この辺から適当な特殊兵器を装備するといいかもしれません。ステージは全部で10ありますが、コンティニューはステージ6まで。頑張って最終面を目指しましょう。

最後に

確かにデキは「まずまず」だし、ゲームも楽しいのですけれど、どうもキャラクターが……。敵キャラはなんかデコレーションゲーキみたいにノッペリしているし、特に女性自キャラはせっかく「人間」なのに「スペースハリアー」の主人公のような人間臭さを感じられないのが少し残念。

お相手は、約1年ぶりの再登場の吉田（みんな覚えてないだろうーな）賢司でした。



面をクリアするとこんな感じで次のエリアへ……



4面のボス、ホーミングをよけるのもヤッカイだ

技術を感じるプログラミング

「ニューラルギア」は、なかなか「技術力」を感じさせてくれます。まず、ディスクアクセス。オプション兵器選択時に、どうも並行してディスクを読んでいるようなのです。うーむ、まるでバックグラウンド処理のよう。あと、面クリア時のフェードアウトの仕方の美しさ！ 写真を見てもえればわかると思うのですが、背景や自機の動きをそのままに、エコーをかけてホワイトアウトしていくんです。うーん、この開発メンバーには、今度はアーケードゲームかな

んかを移植してもらいたいな、ディスプレイ縦置きモードかなんかもついたら奴を（思い始めるとどこまで知らない吉田賢司）。期待しているのががんばってください。

7段階評価

ゲーム性	★★★★☆
操作性	★★★★☆
BGM	★★★★☆
グラフィック	★★★★☆
技術力	★★★★☆

早かったじゃん, 完結編



Komura Satoshi

古村 聡

サコムのノベルウェア、闇の血族の完結編がほぼ予定どおり発売、続編にはなかなかお早い登場でありますね。今回の舞台は予告どおりメキシコシティ。そして、そこでも魅由の周りで奇怪な事件が……。

舞台はメキシコシティへ

1990年初夏、新宿。ひとりのファッションモデルが変死体で見つかった。そしてまたひとり犠牲者が……。殺された2人と同じスタジオに所属していた魅由は、彼女らの死に不審を抱き独自に調査を始めた。そして魅由はその答えを求め親友の里沙とメキシコシティへと飛び立ったのだった……。

里沙は疲れているらしい。どこか抜けているような、変な虚脱感に包まれている。

「怖い……のよ——」

里沙が小さくつぶやいた。

そして里沙が消えた。魅由はシャワーを浴びていた。バスルームの床が一面血の色になった。シャワーから血が!? 直後、里沙の悲鳴。あわてて魅由が部屋に戻ると、もうそこに里沙の姿はなかった……。

魅由はティオティワカンの遺跡にある太陽のピラミッドにきた。影のない南中時のピラミッド。音が消えた。人々のざわめき、喧騒が。人々は背後にいる。音だけの世界。いや、そこは自分以外の音のない世界、だった。そしてそこに魅由は信じられないものを見た。そこに現れたのは……!



X68000用 5"2HD版4枚組 8,800円(税別)
システムサコム ☎03(635)7609

さて、ゲーム周りは?

こんな感じで話は進んでいくわけですが、2、3感想など。まずシナリオについて。前編とくらべるとずいぶんとコ○ル調というか新○子モドキ色というのか、

「はふ、ちょっとためいき」

「んーっもう、JESUS」

はなく……はないですが少なくなった(私が慣れてきてしまったただけだろうか!? 考えたくないなあ)ような気がします。

さて、システム。この闇の血族もほかのノベルウェアシリーズと同じようなシステムなのですが、ノベルウェアもプログラマ的にはずいぶんといろいろ改良されているようで、特に音楽はさすがサコムという出来。いやあ、MT-32を使うとほとんど映画音楽だもんなあ。うーん、はりいうっど。

それと画面効果。こいつはっ、前編では拡大縮小なんかやってくれちゃって、へーてなもんだったのですが、完結編はもっ

とびびりますよお。なーんとあの、

ラスタースクロール

までやってしまっているのだ!! やってくれるじゃん。アーケードのアクションゲームの移植でもないのに(まあ、ノウハウはあるだろうけど。ノベルウェア以外にもいろいろ作ってるしね) こういうことを平気でやってくれちゃうのには……いやいや、まいりましたっ!

この闇の血族も前編だけの状態に比べると、ずいぶんよくなったのではないかと思います。話がまとまっただけでもね(これが本当のオチついた、なんつって……ああ)。なんだかんたいって、シナリオもクライマックスとあの最後のエンディングで、結構感動する人もいそうだし。まあノベルウェアとしてはなかなかよくできたほうになるのでしょうか(私は「ソフトでハード……」の2がいちばん好きなんです)。



▲お!? これは!?
アニメ効果だったりして

◀ついにメキシコにやってきた魅由

ノベルウェアのたどり着く先は?

まず最初に。これはゲームではなく“ページをめくる必要のないBGM&絵つき小説”です。確かにたまにマウスを使いますが、その行為自体は“ゲームを解くため”ではなく“魅由になるため”のものです。次にシナリオですが、これ自体のセンスはわりといい。いま流行の覚醒ものだし。ただ、魅由の行動と思考で成り立っているだけに、どうも先走りや考えすぎのところが多く、見ているとめくれないページを戻し

たくなる衝動にかられます。さてグラフィック。背景はともかく、肝心のキャラクターの表情がちょっと乏しい。謎を解くためにメキシコまでやってきた行動力のある女の子にしては、曖昧な表情が多すぎます。まあ、それでもシステムの進化や雰囲気作りのよさも手伝って、全体的には合格点といえるでしょう。(出口香)

グラフィック	(10段階評価)	6
音楽		9
センス		8
ノベルウェアシステム進化度		9

今年のトレンドはスポーツするヤンキーだ



泉大介氏に触発されたわけではないが、今月はEOS-1000を衝動買いしてしまった荻窪圭である。本当はEOS-10を買いたかったのだが、まあ、いいや。PIXYというEOS-1000といい、はたから見ると「よくそんな金があるなあ」と思われる方も多いだろうが、金はない。あるのはクレジットカードである。お間違いなきよう。

ああ、EOS-1000って、安くて高機能なのはいいけど、本当に玩具だね。大事なこと以外はとにかくそこらじゅうプラスチックで作ってあって、軽すぎてかえってホールドしにくい。日本ってのは、玩具みたいな製品を作るのは得意だね。

ゲームもそう（いきなり無謀な展開だが）。大作やらスケールの大きいゲームは演出文化や大河文化で先を行くアメリカには勝てないけど、玩具みたいなゲームは本当に得意だ。くにおくんシリーズなんか、その典型。ああいった中身のない単純で簡単なゲームを作るのはうまい。シリアスでかっこいい演出をさせると日本のゲーム業界は（映画業界も）ぼろぼろだけど、コミカルで軽い演出はなかなか。アメコミと少年ジャンプの違いかもしれない。



X68000用 5"2HD版2枚組 8,800円(税別)
シャープ ☎03(260)1161

Ogikubo Kei

荻窪 圭

またまた登場のくにおくんが今度はサッカーに挑戦。もちろんルールはなきに等しい。スナオにアクションとして楽しもう。しかし「不良更生＝スポーツで発散」という図式はゲームにまで浸透するんだな……。

あ、そういえば、先月のスタジオボイスの海外コミック特集を見たのだが、フランスのメビウスって人の絵は凄い。描き込みが凄い、っていうのなら大友克洋も凄いが、このメビウスって人の色使いやセンスも凄い。そのぶん、1コマひとコマのスピード感はないけど、スピード感とノリだけで描かれているような日本のコミックを読み慣れてると、こういうのが新鮮でいいね。

で、ここで紹介するゲームはジャパニーズ丸出しの、スピード感とノリだけが勝負のゲームだ。

* * *

いま知ったのだが、サッカーってのは、アソシエーション・フットボールの略語で、アソシエーションを短縮した語に“er”をつけたものなんだそう。「研究社英和中辞典」のSOCCERを引いたらそう書いてあった。初耳。

さて、サッカーといえば、ワールドカップ。マラドーナなんてどうでもいいけど、私は、あのごついキーパーとブロンドの長髪をなびかせて突進するFWが好きだったんだよな。何が悔しいって、あの長髪FWが決勝戦に出られなかったこと。うーん、気に入らん決勝だった。

で、このサッカーゲームである。間違っても「熱血高校サッカー部」ではないぞ。「熱血高校ドッジボール部 サッカー編」だ。ドッジボール部の部員がサッカー部マネージャーの“みさこ”に無理やりくどかれてサッカーするという、いわば、他流試合なのだ。ドッジボール部だって、美人マネージャーの色香にはへろへろなのだ。私だって色香には弱い。絶世の美女に色気で迫られたら、一太郎だって使っちゃうぞ。

さて、このみさこってのがまた曲者なのだ。なんてたって、「私の（ピーッ）なんてどうでもいいのね」なんていって、女を武器にドッジボール部

にサッカーさせようっていうのだから。ちなみに、ピーッって書いたのは、よい子のOh!Xには書けないような言葉だからではなく、本当にピーッっていうからだ。なんか、サンデーやマガジンに出てきそうな女マネージャーだ。

オフサイド、な〜んやそれ？

他流試合とはいえ、ドッジボールより面白く遊べた私である。なんでか。

ドッジボールは、画面上で一度に動くキャラクターがわんさかいるから、ときどき処理が重いのか反応が鈍いときがあったが、サッカー編ではそんなことはない。なんてたって、このサッカー、室内サッカーでもあるまいに、6人制なのだ。キーパーとDF2人と、MF1人と、FWが2人だけなのだ。おうおう、これは楽しい。そもそもって、動かせるのはMF（ミッドフィールダー）だけ。ポジションは変更できるから、くにおくんじゃなくてもいいわけだ。

そもそもサッカーで重要なのは個人技だけではなく、連携プレーであり（たとえば、翼くんと岬くん、薬丸とシズミみたいな関係だ）、作戦だ。それがだな、パス主体で攻めるかドリブル主体か、タックルするか、自分のキャラ以外にシュートをさせてやるかどうかなんて試合前とハーフタイムに設定できるのだ。そのうえ、自キャラがボールを持っていないときには、“パスしろ”だ



くにおドリブルするの図

の“シュートを決めろ”だのの指示は出せるし、相手ボールのときには“さっさとタックルしやがれ”なんていえるのだ。これは楽しい。

味方の動きのアホさ加減にうんざりしたときは指示を出せ、だ。

ほら、なんか、まともなサッカーゲームのような気がしてきたでしょ。まともなサッカーゲームの素質はもってるよ。しかし、痩せても枯れてもくにおくんなのである。例によって人知も物理法則も越えた必殺シュートはあるわ、オフサイドはないわ、がんがんタックルしてもイエローカードは出ないわ（それ以前にファールがない）という超アバウトなサッカーなのだ。

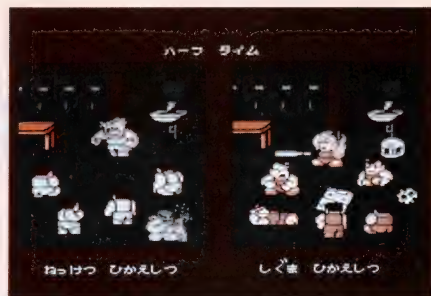
演技の下手なみさことは

まあ、例によって、脳天気なストーリーがある。サッカー部員が食中毒になったから代わりに出場してくれっていうことだ。みんな、いやがってるんだが、みさこの「ピーッしてあげちゃう」のひとりでみんなの目ん玉が「抜作先生」になってしまう（古いなあ）ところもありがちだが、これがまた本当にPCMでしゃべっちゃうところなんか「闇の血族」より凄いぞ。

で、だな、ときどき食中毒の治ったサッカー部員が復帰してきたりする。これがまた、目立っていい動きをするでもない気安め野郎ばかりなのだ。でも、途中でメンバーを変えられるというのはルーチンワークと化しがちな展開に刺激を与えていいぞ。

それでまたいちいちしゃべるんだ。PCで。これがまた、下手くそで下手くそで。プロの台詞棒読み屋とみた。こんな味は素人にはなかなか出せない。なかなか味わい深いので、1度はちゃんと聞いてやろう。みさこのタカビーぶりも腹が立っていいぞ。

試合は至極簡単で、至極面白い。ホイッスルで試合が始まる。まずパスし、くにおくんは指示を出しながら、FWの2人にボールを持ち込んでもらうのがいいだろう。レバーを倒すだけでドリブルしてくれる



▲ハーフタイムにうんこすわりの図

◀ダイビングヘッドシュートの図

し、適当にBボタンを押せば適当な味方にパスするし（適当なぶん、よく、インターセプトされるけど）、Aボタンを押すと、シュートしてくれるので簡単。傑作がA+B。走りながらパスされるボールに合わせてA+Bを押して、ダイビングヘッドバット！ じゃなかった、ダイビングヘッドシュートなのだ。これがまた、必殺「なめんなよ（つてしゃべるんだもん）」が出やすいので重宝だが、タイミングが難しい。空振りして寝転がってるくにおは情けないぞ。

止まった状態でA+Bだと翼くんばりのオーバーヘッドシュート、これがまた気持ちいい。はずすとただみっともないだけなので、何度もくるくる回りながら、「ジャンプで一す」と馬鹿のふりをしてごまかそう。

基本的にパスを受けるタイミングが結構難しいのだな。ノートラップのポレーシュートもできるが、これも結構難しい。

問題は、敵チームだな。トーナメントモードで遊ぶ。対戦するのは全部で12校だ。そう、12校。考えてみろ。12回も戦わなければ優勝できない大会って、いったい何校参加してるんだ？ もの凄いな数だぞ。

まあ、それは色物アクションだからいいとして、敵だ。4回戦で当たるマダギ学園なんて、凸凹の土地で、石にけつまずいて転びながらの試合だし、8回戦の恐山商業高校なんて、氷の上だぞ。滑る滑る（がんばれ受験生！）。

ハーフタイムショーもなかなか楽しい。吉本工業高校の落語とか、一本釣り水産高

校の釣りとか、なかなか笑える。

この一本釣り高校ってのがまた卑怯な必殺シュートを使ってくれるんだ。なんてったって、サッカーボールがおサカナになっちゃうんだ。いくら優秀なキーパーでもかつおは取れないぜ。

攻略法？ あつたら教えてもらいたいのんだ。ゴールキックを蹴ろうとするキーパーの前でゴールキックを止めて（間抜けなキーパーなのだ）、そいつを蹴り込むとか（たまに成功する）、味方にシュートを打たせて、キーパーのはじいたこぼれ球を蹴り込むとかいろいろ手はある。しかし、準決勝の服部学園には通用しないのだ。いくら忍者だからといって、石にけつまずいても転ばないとか、足が異様に速くて追いつけないとか、簡単に必殺シュートを打ちやがるとか、とにかく気に入らないのだ。

2人プレイもあるでよ

例によって2人プレイもある。対戦もできるが、ここはやはり力を合わせてトーナメントを勝ち進んでいきたい。2人プレイだとプレイヤー1がMFを、プレイヤー2がFWのひとりを担当するので、ちゃんと練習すれば息のあった攻撃が可能だ。互いに罵倒しあいながらゲームを進める面白さはちゃんとある。

くにおくんシリーズは、ゲーム専用機の世界では“1本でいろんなスポーツが遊べる”やつもあるらしい。そういうので気楽にお遊びできるやつも出してもらいたいね。

総評

とにかくくにおくんシリーズである。操作の反応はいいし、動きもいいし、コンピュータは（味方も敵も）馬鹿だし、というコミカルアクションの王道を行くソフトである。

飽きたら、ポジションを変えて、くにおをキーパーにしてみたり、変な奴を担当したりして楽しめる。一応、みんな、必殺シュートを持っているので、それを見出す楽しみもある。

こういうゲームばかりになっても困るけど、

アクションゲームが得意ではない私でも1本は揃えておきたいソフトだ。ドッジボールよりおすすめ。V'BALLよりおすすめ。ってとこだな。ワールドコートよりおすすめ……とはいえない。

5段階評価だがや

キャラの反応度	5
キャラの頭のよさ度	3
みさこのタカビー度	5
学芸会度	5
暇潰し度	5



みさこに哀願されるの図

純国産本格派模擬飛行ゲーム



Nishikawa Zenji

西川 善司

特異な前進翼構造を持つ、特殊戦術戦闘攻撃機“MI-C.A.D.O.”。この戦闘機を操り、空中戦、軍事施設への対地攻撃、水上艦隊への対艦攻撃、さらには偵察、艦船護衛任務などのミッションを遂行せよ。

一ズの底力を見せてくれるのか、遊撃王II!?

Let's fly! (道の真ん中で大声で言った
ら実に恥ずかしいセリフだな、これって)

洗練された操作系

まず、ジャパニーズにフライトシミュレータのウケが悪いのは、おそらくその操作系の複雑さに起因しているのだろう。舶来のフライトシミュレータは機能が多いため、日本語ワープロ以上の操作キーがキーボード上に割り当てられており、はっきりいってよほどこの筋のゲームが好きでないとやる気が失せてくる。

たとえば、車輪の出し入れ。離陸したあとも車輪を出していると速度が出ない、燃料効率が悪い、挙句の果ては車輪が壊れてしまうなど（よく言えば）細かいところまで凝っている。こういった面倒臭い操作をアメリカ人はきつと「きゃあきゃあ」いながら嬉しがるんだろう。

で、「遊撃王II」は、というところといった面倒臭い操作は一切排除されているから、この筋が初めての人も安心安心。一度飛べ上がったプレイヤーは空中戦に集中してればいい。よほど変な行為をしない限り「失速」もしないから宙返りや垂直上昇なんかをやってもOK。

そうそう、もちろんサイバースティックにもバッチリ対応しているぞ。レーダーレンジの切り替え以外はキーボードを必要としないから、ほとんどこれひとつで遊べてしまうのだ。ちなみに私は全20ステージをサイバースティックでクリアした。

2つのモードで楽しさ倍増

「遊撃王II」は大きく分けると2つのモードで構成されている。「ミッションモード」と「フライトモード」の2つだ。

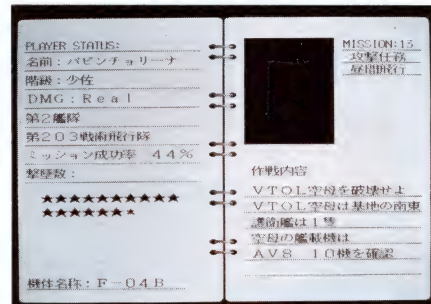
「ミッションモード」は今回の主役である近未来架空戦闘機“F-04B MI-C.A.D.O.”の開発された西暦20××年を舞台にしたリアルなミッション遂行型のゲームだ。

与えられた作戦をうまく遂行できればその功績に応じた階級と新たな任務がプレイヤーに与えられる。

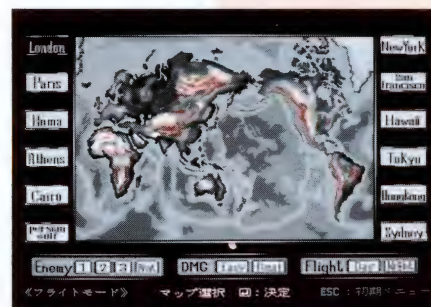
「フライトモード」は“F-04B MI-C.A.D.O.”で、世界中の好きな場所の好きな時間帯のフライトを楽しめるモードだ。こちらは敵機の有無や攻撃を受けたときの操作系統への影響などを自由に設定することができるので「ミッションモード」をする前の飛行訓練にもいいぞ。

また、ロンドンでは「タワーブリッジ」の下をくぐったり、ニューヨークでは「自由の女神（らしきもの）」の上空を飛んだりして、もう気分は世界一周旅行だ。

さて、「ミッションモード」では「ユーザーディスク」なるものを作成しなければならない。RPGのようにゲームの途中経過をこのディスクに記録することができるわけだがひとつ注意しなければならないのがダメージモードの設定だ。これは敵機からの攻撃を受けたときのダメージの操作系統への影響の有無を設定するもの。前述の「フ



ミッションの内容はよく読んでから飛べ



好きな国の好きな町をここで選ぶ

アメリカのコンピュータ野郎は、そういうフライトシミュレータ好きらしい。

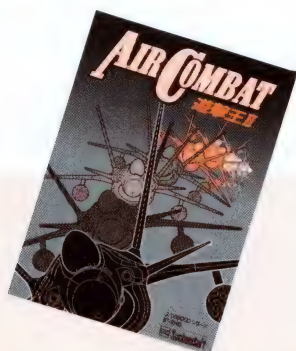
「パソコンを買ったら作ってみたいゲームは？」

なんて質問をアメリカのパソコンショップでAMIGAなんかに齧りついてる小僧に浴びせたら、十中八九が「フライトシミュレータさ、ペイビー、ファクキューメン」とか答えるんだろう。

さて、もしこれを我が国ジャパノンの人間に聞いたら、何と答えるんだろうね。口を揃えたように、「〇ースミたいなアクションRPG」とか「〇ラデ〇ウスミたいなシューティングゲーム」なんていう答えが返ってくるのが目に見える……。

好きだというだけあって、舶来フライトシミュレータはよくできている。画面がガシガシウニウニ動くのなんの。視点は変えられるわ、道に車は走ってるわ(F-29)、オートパイロットが賢いわで(Falcon)、パソコンで動いているとは信じたくないほどの凄さ。

さて、今回発売された「遊撃王II」。アメリカのフライトシミュレータをかなり意識した作りになっている。さあ、ジャパニ



X68000用 5" 2HD版2枚組 8,800円(税別)
システムソフト ☎092(752)5278

ライトモード」ではプレイするたびに自由に設定できたが「ミッションモード」ではプレイヤー一名登録時で一度設定してしまうと途中で変更ができないのだ。

これは純粋に通常のゲームの「EASY」と「HARD」のようなランク設定に相当するので自分の腕前にあわせて慎重に選ぶように。私は「REAL」（つまりは「HARD」）をお勧めするぞ。ダメージを受けるたびに操作系

統がイカれていくのがリアルだし、この状態で無事に帰還したときの充実感といったら、スポロガムの絵をうまく切り出せたときの快感に通じるものがあるぞ。

多彩な内容の各ミッション

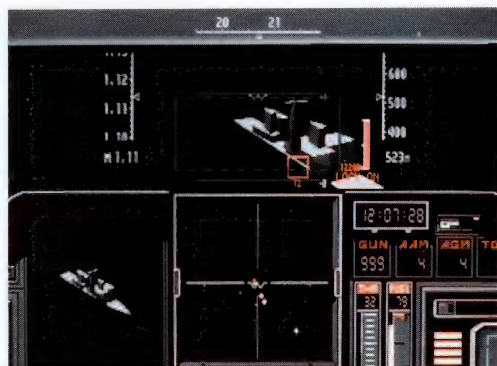
ミッション1はオーソドックスな「敵機を全機破壊せよ」といった内容だ。単純な内容とは裏腹にミッション1は実質的に一番難しい作戦なので、ここでもう断念してしまう人も多いのではないだろうか（破壊すべき敵機の数8機、自機のみサイル搭載数は4発。つまり、最低でも4機は機銃で撃ち落とさなくてはならないのだ）。でも、あきらめてはいけない。このゲームが面白くなってくるのはミッション2以降なのだ。

その先のミッションを紹介しておこう。ミッション2は敵基地の偵察任務。これは敵領空へ低空より侵入し、敵基地の写真を撮影し帰還するといった作戦だ。ミッションは敵機を撃ち落とすだけでなく、こういったリアルな内容のものがある。しかも夜間飛行。[R]キーまたはサイバースティックのセレクトボタンで画面をワイヤーフレームに切り替え可能なのだが、これが暗視カメラみたいなノリをかもしだして、なかなかグーな演出。

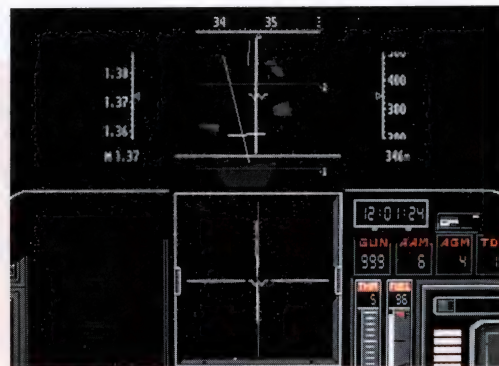
そして、次のミッション3はミッション2で偵察した敵基地の破壊作戦。うーん。知らず知らずのうちにプレイヤーを引き込



これが護衛すべき機体。あとでこいつが……



敵艦隊発見、対地ミサイルで攻撃だ！



着艦寸前！緊張する一瞬だ

んでいく、言葉で語らずして張られた見事なこの伏線。

さて、この先にも多種多様な内容のミッションが用意されているぞ。たとえば、味方輸送機の護衛任務とか、敵艦隊の殲滅、自機に優るとも劣らない速さで飛来する敵新型戦闘機の撃墜などなど。まあ、あとは自分の目で確かめてくれや、健闘を祈る！

着陸、着艦のテクニック

というわけで、無事にミッションを果たせば基地に戻らなければならないのだが、結構着陸が難しい。せっかく作戦内容を果たしても、ちゃんと基地に帰還できなければ元も子もあつたもんじゃない。そこで、私、バビンチョ西川の着陸テクニックを伝授しよう。

まず、当たり前だがランディングモードにする。ピーコンを参考に適当な距離を残して機体を滑走路に平行にする。速度は800km/h以下、高度は600m以下にして機首をやや下に向ける。まずは下準備といったところだ。

滑走路が見た目にかなり大きくなってきたら速度400km/h以下、高度は300m以下にする。そして、滑走路へ進入しそうになるころには速度200km/h、高度12~20mになるようにする。

滑走路へ進入したら機首は水平くらいに

合わせる。あとはエアブレーキをかけて一気に速度を落とす。そう、高度を操縦桿でコントロールせず、速度を落とすことによって失速させ高度を下げるのだ。この方法は、特に走行距離の短い空母へ帰還する場合に有効だぞ。なぜなら着艦したときにはすでに速度が十分落ちているため短い制動距離で停止できるからだ。

気になる処理速度は？

このテのゲームで一番気になる点が処理速度だ。「遊撃王II」は残念ながらお世辞にも処理速度が速いとはいえない。ワイヤーフレームモードにすると若干速く感じるがそれでも十分とはいえない。256×256ドット画面などの低解像度モードを使用するなどしてももう少し高速化を図ってほしかった。

また、動きが多少粗い。DoGAのCGAやアメリカのフライトシミュレータを見てもわかるが、このテの3Dものは絵の解像度よりも動きの細かさとスピード感が重要なのだ。

気になる点は本当にそのくらいであとはゲームバランス、敵の賢さ、操作性、グラフィック、どれも非の打ちどころのない出来となっている。ゲームの面白さは私が保証するからぜひ買ってじっくりと遊んでみてほしいな。

PC-9801版との相違点

- 1) まず、第1にレーダーが格段に見やすくなった。PC-9801版では大地図の上に自機や敵機を表す点がポツポツと動くだけだったがX68000版では表示を拡大縮小できるジャイロコンパスのようなものになり、敵や滑走路までの距離などを把握しやすくなった。
- 2) 第2に画面左下に敵機映像が表示されるようになった。これは自機の視点から見たもので、これによって敵機との相対的な位置を瞬時に判断することができるようになった。
- 3) 第3に視点の変更が可能となったこと。カーソルキーで4方向に視点を変えることができる。2)とあわせて使えば敵機を効率よく追っ

とができるぞ。

- 4) アナログスティックでの操作が多少難しくなっている。これはスティックに「遊び」が存在し細かな操作を困難にしているのが原因だ。まあ、これは慣れればたいした問題じゃないかも。

(7段階評価)

ゲーム性	★★★★★★
スピード	★★★★☆☆
サウンド	★★★★☆☆
グラフィック	★★★★☆☆
操作性	★★★★☆☆
お買い得度	★★★★☆☆

きもちいいアクションゲームくださいな

Urakawa Hiroyuki
浦川 博之

トリートンシリーズの悪玉、バルーサの魂が復活した。主人公スティルはそんなこととは知らず、魔物に襲われたエデルの姫ティアの話を聞き、ティアの兄を捜しに出かけた。なんでもいから、剣をびゅんびゅん振って、先に進め！



ある日のソフトショップ「うりやかわ」いらっしやいませ。ゲームをお求めですか。ああ、アクションゲームを買いに。そうですね、「サイバリオン」なんかもいいけれど、私のお勧めはこの「バルーサの復讐」かな。サインソフトの新作だ。おや、どうしたんだい、いきなり泣き出して。よし、それならゲームを見せてあげよう。またまたサイドビューのアクションゲームだけど、キャラクターも大きいし、動きも複雑だよ。演出も凝ってる。まあ、そういうわけで。テストプレイさせてあげるから座って座って。

展示発表試遊会

まず、デモから見るかい。はい、デモディスクとAディスク。プレイのときはAとBのディスクを入れれば、いちいちデモを見なくていいから便利だよ。それにこのデモは512×512モードを使ってるんだ。枚数は少ないけど絵もきれいだし字も読みやすいだろう。

デモが終わったらスペースキーだ。「ディスクを入れ替えてリセットしてください」だってさ。リセットするのが情けなく

ていやだ？ 細かいところにこだわるねえ。遊ぶ前からそう文句をいうもんじゃないよ。じゃ、リセットするよ。

ほら、(しゅわっ) このメニューの字の出方がカッコいいだろう。拡大縮小するんだ。コンフィギュレーションで(しゅわっ)ディスプレイモードや難易度も選べる。ゲームスタートを選んで。

ほら、1面は船の上だ。サンプリングの嵐の音がいいだろう。揺れる船！ 逆巻く波！ BGMはユーロビート調だ。早くゾンビを倒して。しゃがみ突き、ジャンプ斬り、ジャンプしながらの下突き攻撃もできるよ。下突きがいちばん強力だ。おお、うまいうまい。きれいに乗り切った。ここでいきなりボスキャラ大ガニ君が登場だ。沸きだすガニ君の攻撃にめげてちゃいけない。大ガニ君の目玉を狙って下突きた。ああ、なかなか早く倒したね。1面クリアだ。ここでビジュアルシーンがあるよ。

……うふふ。このビジュアルシーンがまたザインらしいんだ。ははは。あははは。君だって、あはは、ウケてるじゃないか。このセンスがさすがザインだろう。

2面は街中。蛙スライムや怪物剣士が相手だ。あ、その灰色の玉は体力ダウンアイテムだから取っちゃだめだよ。ああ、だめだっていつてるのに。タイミング的に取ってしまいやすいんだけど、この敵はこのアイテムというパターンを覚えれば大丈夫。特殊アイテムや新しい武器も手に入るからいいじゃないか。さっきのサルが出てきたよ。ここは謎を解かなくてはならないんだ。ビジュアルシーンを見たからわかるだろう。そうそう。

じゃ、もういいかい(プチ)。

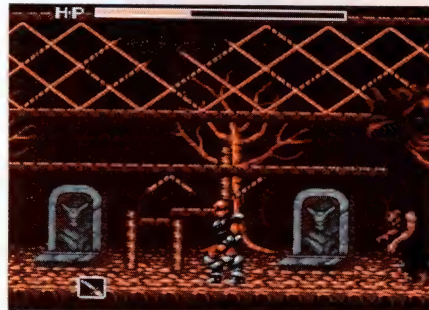
毎度あり

どうだい、面白かっただろう。まだまだこの先、奇想天外な展開があるよ。隠れステージもあるし。まあ、ちょっとキャラクターの動きや背景の絵の作りがぎこちないところはあるけど。そうだね、自分の武器

の威力ももうちょっと強いほうがいいかもしれない。しかし、マップの構造も考えてあるし、敵をガンガンやっつけるっていうアクションの基本的な面白さでは十分合格だろう。いろんな武器のどれが有利か見ていくだけでも楽しいもんだよ。さあ、どうする？ 買うの。そうか、ゲーム中は結構楽しんだものね。はい、じゃあこれ。どうもありがとう。またどうぞ。



こんなカニとも知り合いとは魔王も顔が広い



ゾンビと亡霊の複合アタック

店主うりやかわ氏のメモ

アクションゲームとしての仕上がりは合格点。下突きのしがし敵を倒すのが気持ちいいし、ザインならではのデカキャラも魅力。アーケードからの移植などに比べるとまだまだあちこちで見劣りがするが、個性がその分をカバーしている。なにか(いい意味での)B級映画のような不思議な魅力のあるゲームだ。

操作性	7
ゲームバランス	5
音楽	8
グラフィック	5
ザインのセンス	10
熱中度	8

(10段階評価)



X68000用 5"2HD版3枚組 8,800円(税別)
サインソフト ☎078(242)2855

スプレッドシート®の精神

Ogikubo Kei 荻窪 圭

「大人のためのX68000」も3回目になる。そろそろ本稿が「仕事のためのX68000」でも「企業のためのX68000」でもないことを意識していただけるのではないかとと思う。パソコン自体を楽しむのもまたおつだが、パソコンを使ってする行為を楽しむのもまた面白いもの。

そう考えると、日本はまだまだ(X68000に限らず)遅れているように感じる。パソコン自体を楽しむアマチュアプログラマやマニアと呼ばれる人々と、仕事のためにパソコンを使うビジネス屋さんはたくさんいるが、パソコン文化を担うべき「パソコンを使って何かしてみる」人々の盛り上がりがいまいちなのだ。

だから、市販ソフトをうまく使って、その設計者が「あ、こんな使い方もできたのか」と驚くようなことをしている読者がいたら、ご一報いただけるとうれしい。

* * *

本場アメリカでは表計算ソフトとはいわない。スプレッドシートという。このあたりが非常にアメリカンである。スプレッドというのはSPREADと書き、広がるとか広げるといった意味の動詞である(パンに塗るものという意味もある)。シートは1枚の紙だから、スプレッドシートというのはそのまま訳すと、「広がった紙1枚」ということになる。

つまり、目の前にあるのは「表計算」するためのものではなく、ただの「広がったシート」にすぎないのだ。そのとおり、スプレッドシートは計算にとらわれず、好きなように使ってよいたくさんの升目にすぎないのだ。このあたりを押さえておかないと、「表計算ソフトはビジネスソフトであるから、普通の人には関係ない」といった短絡的な思考に陥りかねない。日本語訳が悪いために意味が限定されてしまった一例だ。

私にとってはスプレッドシートもゲームも同じX68000上のアプリケーションにすぎない。

1. 銀行と小金とせこい利息かせぎ

などといいつつ、実にスプレッドシートらしいお金の計算なんてしてみようかと思う。真剣にやろうと思うと小数点以下の処理や大きな値を扱うときの誤差なんかも無視はできないが、私は銀行屋ではないのでそこまでは考えない。

老若男女問わず、我々は銀行なり郵便局なりにお金を代表とする資産(大袈裟な言葉だこと!)を預けている。貸しているといってもいい。銀行はその集めた金をまたほかのところへ貸して高い利子を取り、我々にはそこから手数料を引いた安い利息をつけてくれるわけである。

で、銀行といっても日銀から第一勧銀から街の農協までいろいろある。まず日銀(日本銀行)。こいつは銀行の総元締めで、全国の銀行は日銀から金を借りたり、日銀に金を預けたりしている。日銀はおいておいて、その他、銀行には以下の種類がある。都市銀行、地方銀行、相互銀行、長期信用銀行、信託銀行、信用金庫、信用共同組合、労働金庫、農協という感じだ。いや、べつに銀行講座をやるつもりはない。

そのほか、郵便局と証券会社も忘れてはいけない。あと、地方銀行と相互銀行から鞍替えした第二地銀を分けるのもいいかもしれない。

でもって、お金がとびかっているわけだが、たいていの場合、磁気ディスク上のデータが行ったり来たりしているに過ぎない。しかも大きな金額を動かしているのは企業が中心であるから我々庶民にはあまり縁のない話ではあるな。

今回は「預けて増やさせる」ということを中心に考えて計算をしてみようと思う。我々が預金できるのはたいした金額ではないので、使える預金の種類も少ない。また、我々庶民の手の届くような預金の金利(つまり、利率)というのは規制金利といって利率が決まっているので、どの銀行だと

なにかにつけKamikazeを立ち上げるという荻窪氏。画面に広がる1枚の紙は心掛けしだい自由なワークスペースとなる。表集計=会計といった堅苦しい事務用途のイメージにとらわれず、スプレッドシートの醍醐味を味わってみたい。

得だということもない。

まず、固定金利にするか変動金利にするかだ。が、何がお得かという話などする気は毛頭ないし、ましてやこれはOh!Xである。間違ってもその辺の週刊誌のマナー講座と一緒にしないように。

さて、固定金利っていうとお金を預けた時点の金利が満期日まで変わらないものである。普通銀行が扱う預金は基本的に固定金利だ。普通預金ってのは例外。

変動金利というのは金利の変動がそのとき預けていたお金にそのまま反映するものである。たいてい半年複利っていう計算をしているので、半年ごとに利息がでる。そのとき、次の半年間の金利(利率)にその時点での金利が反映するのだ。信託銀行が扱っている商品(商品ってのもピンとこないが)が基本的に変動金利だ。

●金利と利息と税金と金融商品

具体的にいこう。

善良な市民は銀行に「普通預金」の口座というのを持っていて、そこに給料が振り込まれたり、そこから丸井やセゾンのクレジットを払ったり、株の下がったNTTやら原発の好きな東京電力やらが定期的に金を引き出していく。まあ、財布みたいなものだ。だから、利率も微々たるもので、雀の涙みたいで、銀行のほうも普通預金が多くてもあまりおいしくない。

で、普通預金にある程度余裕ができてくると、もっと利率がいい手段に訴えたいようになってくる。銀行の女の人もやたらいろいろと勧める。代表的なのが普通銀行の場合、定期預金(期日指定定期)か定期積立で預金だ。これが100万円以上あるなら、小口MMCっていう新しい技もある。

ここで貪欲な人は普通銀行以外にも手口があることを知る。信託銀行へ行けば変動金利で一見お得なヒット(金銭信託の一種)やビッグ(貸付信託の一種)が、証券会社へ行けば中国ファンドがあるわけだ。こんなところがポピュラーで、元金保証がなくてかえって損をする覚悟があるなら投資信

託っていうのもある。

そこで4年前。昭和61年の6月になけなしのボーナスから20万円ほど捻出して2年間預けたとしよう。どうして4年前なのかはおおいわかることである。

選択肢は先に挙げたようにいろいろあるが、期間を考えて、普通銀行でポピュラーな期日指定定期、信託銀行でポピュラーなビッグとしてみた。20万円を2年間預けるというとなると、このどちらかが妥当なところだ。その利率を見てみよう。

当時、2年ものの定期預金の利率は年4.38%だった。期日指定定期ってやつで2年間預けると1年複利で計算ということになるので、そうする。複利というのは、利息が出るたびに、元金にその利息を足したものに対して次の利息が計算されるというもの。

2年ものの貸付信託の予想配当率は年4.58%だった。予想配当率というのは、絶対その利率で利息が返せるとは限らないよ、という意味なのだが、実質的に利率と考えて間違いなさそうである。

ビッグってのは半年複利型の貸付信託である。信託ってのはまあ、信託銀行が扱うことのできるもので、変動金利の金融商品だと思っておけばいいだろう。詳しく話すと長くなるからやめておく。

さあ、Kamikazeである。

まず、期日指定定期だ。これは固定金利のために計算は簡単。

$$20\text{万円} \times 1.0438^2$$

$$(\text{元金} \times (1 + \text{利率}/100)^{\text{年数}})$$

である。いちおうスプレッドシートらしく預金額と利率は別のセルに入れておいた。なお、この式は1年複利である。2乗しているのは利息計算が2年で2回出る複利計算だからである。

続いてビッグだ。これは変動金利である。ビッグは半年複利。変動金利の半年複利。これは半年ごとに利息を出して利率を見直す。計算するときは半年ごとに(利率+1)を掛けていってやればいいわけだ。年利回りが4%なら半年当たり2%となる。計算式は、

$$\text{元金} \times (1 + \text{最初の利率}/2/100) \times (1 + \text{次の利率}/2/100) \times \dots$$

である。これが図1の解説だ。

昭和61年といえば、公定歩合が低く低くなっていった時期であり、それに影響を受けて金利も安かった(金利が低く景気がよくなってきたために、株があればよかったのだともいえる)。そのあたりの事情は図2のグラフにある。Kamikazeのグラフ作成の能力を越えたものになってしまった

(横軸の目盛りが細かいため、うまく表示されなかったようだ)ので、少々見苦しい。

公定歩合ってのは、日銀が民間の銀行に対してお金を貸し出すときの利率である。こいつの上下が金利や景気に影響を及ぼす。歩合ってのは割合のこと、公ってのは御上のことだから、御上の定めた割合って意味だと思えばいい。景気をよくしたいときは下げ(安い金利で金を借りられるので、金を借りてことを起こす連中が増える)、景気がいいときは上げるのが普通らしい。

で、結果として変動金利のビッグが損をしている。これは珍しいケースで、定期預金より利率の高いビッグでもこういうことはあるのだ。

しかし、公定歩合が上がり始める63年12月に預けたとしたらどうだろう。

というわけで、結果が図3である。図1とあわせて見れば銀行に金を預けるということがどういうことか少しはわかるだろう。

あと、昭和63年から導入されたマル優廃止による“利息の20%は国のもの”という乱暴な政策により、結果として受け取れる金は減ったりする。ああ、前門のマル優廃

止、後門の消費税というわけで、このときより貯めても使っても税金がかかるようになったのだ。いくらなんでも2割は多いが、閑話休題。

ついでだから、平成2年10月から2年間預ける場合、金利は変わらないとして計算したのが図4である。

新しく飛び込んできた中国ファンド(中期国債ファンド)といって、証券会社はその30%以上を中期国債で運用することに法律で決められている)は1カ月複利であるから、結果としてお得なように見えるが、税引き後を見ると、ビッグより安い。

これは、中国ファンドが1カ月ごとに20%の税金を取られるのに対し、ビッグは満期時の利息に対して税金がかかるようになっているためである。まあ、満期まで取り出せない(中途解約すると手数料を取られる)ビッグに比べ、いつでも引き出せる中国ファンドの魅力はある。

利息にかかる税金の計算であるが、ビッグのように満期時に一気にかかる場合は、満期時に受け取るはずの金額から元金を引いて(これが利息ね)それに0.8を掛けて

図1

昭和61年6月末の場合

定期預金2年間の場合			
元金		¥200,000	
利回り		4.38%	
預入期間		2	
満期時		¥217,904	

ビッグ2年間の場合			
元金		¥200,000	
預入期間		2	
利回り		61/06 4.58%	
		61/12 4.21%	
		62/06 3.84%	
		62/12 3.84%	
満期時		¥216,985	

図3

昭和63年12月末の場合

定期預金2年間の場合			
元金		¥200,000	
利回り		3.64%	
預入期間		2	
満期時		¥214,825	
税引後		¥211,868	

ビッグ2年間の場合			
元金		¥200,000	
預入期間		2	
利回り		63/12 3.84%	
		1/06 4.40%	
		1/12 4.77%	
		2/06 6.08%	
満期時		¥219,777	
税引後		¥215,822	

図2

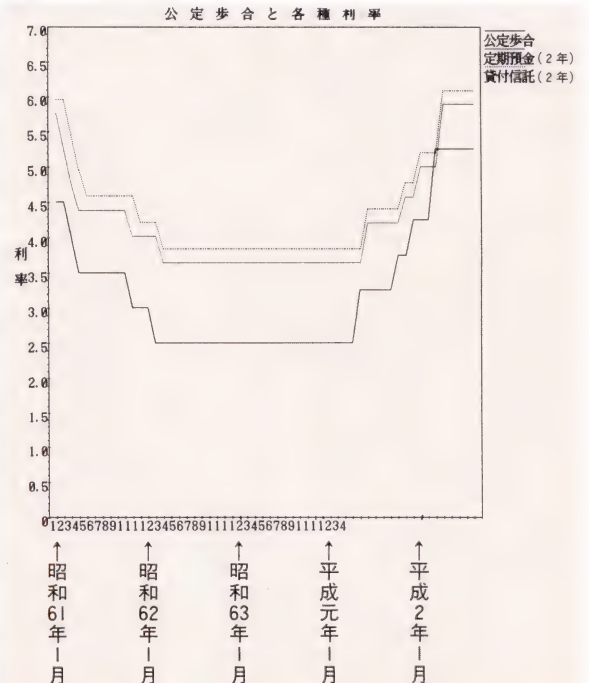


図4

平成2年10月15日現在の利率による計算

元金 ¥200,000			
期間 2			
	利率	2年後	税引後
期日指定定期(2年)	6.33%	¥226,121	¥220,897
ビッグ(2年)	6.53%	¥227,427	¥221,942
中国ファンド	6.497%	¥227,672	¥221,868

やれば税引き後の利息はわかる。中国ファンドのようにその都度税金を取られる場合は、あらかじめ利率に0.8を掛けて計算すればいい。

利率の差なんて2年程度ではそんなに違うものではないね。これが5年くらいになると差は開いてくるけれど。まあ、計算してみると面白いだろう。5年ものの貸付信託の利率はいまのところ、年8.02%である。一般に1年複利より半年複利が、それより1カ月複利がお得なことになっている。

今のご時世、明日は明日の風が吹くので、何が得かといわれても、困るが、少なくともKamikazeでも使って計算してみれば、一見何が得かわからないような数字に関しても具体的に判断できていいこともある。

ちなみに、非常にお得な大口定期という自由金利商品もあるのだが、これは最低預け入れ額が1000万円なので、考えもしなかった。いつかは大口定期を使える身分になってみたいものである。

ちなみに、こういった利率は新聞（朝日新聞なら週末の夕刊）に載っている。

あと、預ける話ばかりしたが、金利が高いということは、預けるときのにも借りるときにもいえることなので、ローンを組む場合は不利である。金利の安いときに固定金利でローンを組んだ人は、結構お得感があるだろう。Kamikazeの財務関数を使えば借った金を返済する計画を立てる助けにもなるが、とりあえずそういう縁起のよくなさそうな話は避けておこう。

2. 売上管理のおもちゃ的例題

スプレッドシートというと、日常業務でデータ処理に使うか、ある程度まとまった表を入力してその分析に使うかどちらかというのが一般的のようだ。

意志決定支援（なんて大袈裟な言いぐさ！）に使うのもいいらしい。ひと通りデータを入力したあと、製品を5%引きにして売り上げが10%伸びたとしたら果たしてどのくらいメリットがあるか、ってな計算を

図5

製品コード表		
製品コード	商品名	単価
A3 ← CZ-603C-BK	X68000 EXPERTII	¥338,000
CZ-603C-GY	X68000 EXPERTII	¥338,000
CZ-613C-BK	X68000 EXPERTII-HD	¥448,000
CZ-623C-TN	X68000 SUPER-HD	¥498,000
CZ-653C-BK	X68000 PROII	¥285,000
CZ-653C-GY	X68000 PROII	¥285,000
CZ-663C-BK	X68000 PROII-HD	¥395,000
CZ-663C-GY	X68000 PROII-HD	¥395,000 → C10

させるのである。

まあ、とりあえずそういった使い方に焦点を当ててみる。商品Aなどとやってもつまらないので、今発売されているX68000を表にしてみた（図5）。面倒なので、本体のみである。この表に製品コードという名前をつけてセーブしておく。

さて、売上管理である。記録するときにはなるべく楽をしたい。いちいち商品名を打ち込むより、その製品に付けられた製品コードと数量を打ち込むだけであとはパソコンがよきにはからってくれるというのが望ましい。

そこで、図6である。たとえば、こんな結果になる。一見なんということはないが、実際に打ち込んだのは製品コードと数量だけだ、というのがミソ。

ここではVLOOKUP関数を使ってみた。
=VLOOKUP(B3,製品コード!A3:C10,1)

っていうのは、図5の“製品コード”という名前のシートのA3からC10を対象にして、B3セル（売上表の製品コードが入っているセル）と同じものを探し、その右のセルの値（製品名）を返すというものだ。厳密には同じものを探すわけではなく、値を比較しているだけなので、製品コード表は製品コードで昇順にソートされている必要がある。で、製品コードのシートではA列に製品コードが、B列に商品名が、C列に単価が入っているの、A列のひとつ右は製品名、2つ右は単価になるわけだ。

ここでは製品コードという名前でセーブした図5のシートを製品コードテーブルとしてアクセスしている。そのためには環境設定のパスにそのファイルがなければならぬ。この関数に限らず、Kamikaze君は他シートの参照時にディスクを読みに行くので少々時間がかかるのと同時に、ディスクに最新版をセーブしておく必要がある。

この“他シート参照技”を覚えると、けっこう応用が利くので便利である。普通、データベースなんかを使うとき、いくつかのパターンしかない項目についてはそのコ

図6

10月X68000売上表				
製品	数量	製品名	単価	合計
CZ-603C-BK	23	X68000 EXPERTII	¥338,000	¥7,774,000
CZ-613C-BK	19	X68000 EXPERTII-HD	¥448,000	¥8,512,000
CZ-653C-BK	10	X68000 PROII	¥285,000	¥2,850,000
CZ-653C-GY	18	X68000 PROII	¥285,000	¥5,130,000
CZ-623C-TN	19	X68000 SUPER-HD	¥498,000	¥9,462,000
CZ-663C-GY	8	X68000 PROII-HD	¥395,000	¥3,160,000
合計	97		¥2,249,000	¥36,888,000

ここだけ入力 こっちは自動

ードを設定しておき、コードを入れるだけで済みますのが基本だ。

たとえば、地区コードを47都道府県に割り振っておけば、入力時に、数字を打ち込むだけで県名が出てくるようにできる。いちいち変換するより、数字を打ち込むほうが早い。

蔵書管理の場合、その本の大きさによってコード001は文庫で、002は新書でってやっておくと、入力時に数字を入れるだけで、あとはVLOOKUP関数で“文庫”っていう文字が自動的に入るようにできるのだ。なんか、コンピュータを使っている気分である。

さて、先月、Kamikazeは大きなシートを作ると遅さが際立ってしまって損をした気分になるという話をした。よって、他シート参照を多用すれば、1つひとつの表が小さくても十分活用できるのである。

たとえば、1カ月分の日ごとの明細シートを作り、1カ月分集計した結果のシートを作ってそこに集計し、12カ月分集計したら、年間集計シートを作ってそこに集計するという作業をすることも可能だ。

もちろん、自動再計算はOFFにしておく。いちいちディスクに行くのはとてもうっとうしいので、再計算は手動だ。

3. 集計作業も関数でできてしまう

で、図7である。ちょっと趣向を変えて、X68000の毎日の売り上げを何割引きで売ったかも加味して作ったものである。あくまでもサンプルね。紙面の都合もあって、6日分だけである。ここから、データ範囲関数ってのをを使って、機種ごとの合計を算出した（図8）。データ範囲関数というのは、任意の範囲のデータに対して条件にマッチしたものを対象に行う統計関数である。すごく便利だが、すごく面倒臭い。

たとえば、X68000EXPERTII（ブラック）つまりCZ-603C-BKの合計販売数を求めるのは、

=DSUM(元データ,3,G18:G19)

である。元データっていうのは図7の範囲につけた名前だ。3ってのは、元データの3列目に売った個数が入っているからだ。で、G18とG19にはマッチングの条件が入っている。

G18にはマッチング対象が2列目の製品コードだから2。

G19はCZ-603C-BKを検索するのだから、

=“CZ-603C-BK”

図7

10月X68000売上明細						
日付	製品コード	数量	製品名	割引率	売値	合計
1	CZ-603C-BK	3	X68000 EXPERTII	10%	¥304,200	¥912,600
	CZ-603C-BK	1	X68000 EXPERTII	20%	¥270,400	¥270,400
	CZ-613C-BK	2	X68000 EXPERTII-HD	25%	¥336,000	¥672,000
	CZ-623C-TN	1	X68000 SUPER-HD	18%	¥408,360	¥408,360
	CZ-663C-GY	1	X68000 PROII-HD	30%	¥276,500	¥276,500
2	CZ-653C-GY	2	X68000 PROII	20%	¥228,000	¥456,000
3	CZ-613C-BK	3	X68000 EXPERTII-HD	10%	¥403,200	¥1,209,600
	CZ-653C-GY	2	X68000 PROII	25%	¥213,750	¥427,500
4	CZ-623C-TN	2	X68000 SUPER-HD	32%	¥338,640	¥677,280
	CZ-653C-GY	1	X68000 PROII	25%	¥213,750	¥213,750
	CZ-653C-GY	2	X68000 PROII	20%	¥228,000	¥456,000
5	CZ-613C-BK	3	X68000 EXPERTII-HD	15%	¥380,800	¥1,142,400
6	CZ-603C-BK	5	X68000 EXPERTII	30%	¥236,600	¥1,183,000
元データ						

元データ

である。つまり、元データから製品コードがCZ-603C-BKであるものを引っ張り出して、その販売数を合計しなさい、てな指令になるのだ。なんて便利。スプレッドシートらしい使用法の一例だ。

ただし、これはこういう小さい表だからいいのであって、1日に何十項目も入力のある業務で使って実用になる速度を得られるかどうかの保証はしない。

4. 初歩的な数学とスプレッドシート

風の便りに聞くと、本場アメリカでは科学技術計算などにもスプレッドシートは使われているそうである。なんでも、循環参照機能を使って、初期値を指定した再帰っぽい計算のできるものもあるそうである。

Kamikazeでも簡単な数学系の関数は用意されている。循環参照計算もできる。

循環参照とは、A1セルに $=A2+5$ とあって、A2セルに $=A1+2$ とある場合、A1を計算するにはA2セルの値が必要で、A2セルを計算するにはA1セルが必要という互いに参照しあうものだ。デフォルトでは“こういうことをされては困ります”ということになっているが、解禁することも可能だ。

解禁するとどうなるかというと、無限に参照しあってしまう。いつまでも計算してオーバーフローで止まるか、本当に止まらない。それを防ぐために、反復回数の指定ができるようになっている。

ここで私は思った。うまく収束してくれる関数であれば、簡単に結果が求められるのではないか。が、そうは問屋が卸さない。初期値の指定ができないので、掛け算や割算が入るとうまくいかないのだ。よって、階乗の計算なんか無理だった。残念。

それでは、ということで、連立方程式に挑戦してみた。図9である。

まずセルを2つ使い、式を変形して $X =$

図8

10月X68000集計		
	数量	売り上げ合計
CZ-603C-BK	9	¥2,366,000
CZ-603C-GY	0	¥0
CZ-613C-BK	8	¥3,024,000
CZ-623C-TN	3	¥1,085,640
CZ-653C-BK	0	¥0
CZ-653C-GY	7	¥1,553,250
CZ-663C-BK	0	¥0
CZ-663C-GY	1	¥276,500
合計	28	¥8,305,390

条件1	2	2	2
条件2	_="CZ-603C-BK" _="CZ-603C-GY" _="CZ-613C-BK" _="CZ-623C-TN" _="CZ-653C-BK" _="CZ-653C-GY" _="CZ-663C-BK" _="CZ-663C-GY"		

DSUM関数用条件式

の式と $Y =$ の式にする。それぞれをセルに割り当て、循環参照させる。解に向かって収束していつくれるはずなので、循環回数を増やすほど（Kamikazeの扱える有効数字を限度として）答えが求められるはずだ。

扱った式は、次の3つだ。

$$Y = X^2, Y = X + 2$$

$$Y = X^2, Y = 2X + 2$$

$$Y = 3X + 20, Y = 9X$$

一見うまくいっている。しかし、なんか変だ。連立2次方程式の解は2つあるのだが、図ではひとつしか求まっていない。どこに問題があったのかは一目瞭然。最初の解に向かって収束していくだけなのだ。

具体的には、Xが正である最初の解に向かって収束する。釣り合ったら、そこで同じ値のまま無駄に計算を続ける。負の領域にある解を無理やり求めようとしたら、値が発散して（ $X = 30000$, $Y = -30000$ てな感じ）しまった。

連立1次方程式の場合、素直に答えが出ているように見えるが、それも答えが出るような式を作ったからであって、解が負になったりすると、収束しないで発散してしまう。考えればすぐにわかる。まあ、循環

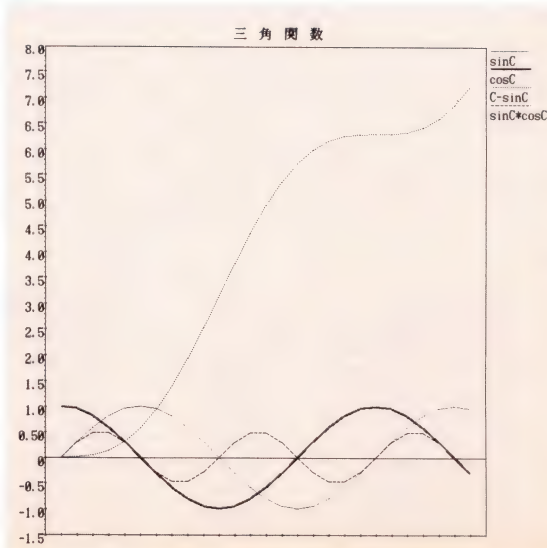
図9

連立方程式を解く（反復回数10回）		
2次方程式	X=	1.9999976469 X=SQRT(Y)
	Y=	3.9999905876 Y=X+2
	X=	2.7318679265 X=SQRT(Y)
	Y=	7.4631023677 Y=2X+2
1次方程式	Y=	29.999491947 Y=3X+20
	X=	3.3331639825 X=Y/9

図10

連立方程式を解く（反復回数40回）		
2次方程式	X=	2 X=SQRT(Y)
	Y=	4 Y=X+2
	X=	2.7320508076 X=SQRT(Y)
	Y=	7.4641016151 Y=2X+2
1次方程式	Y=	30 Y=3X+20
	X=	3.3333333333 X=Y/9

図11



コンピュータシミュレーションの世界

現象をモデル化して状況を分析し新しい状況を予測する。これはもっともコンピュータらしい仕事ではないでしょうか。この連載ではシミュレーションの基本的な考え方から制御言語の作成まで、集中的に対応していく予定です。

Kamon Masato 華門 真人

すべての始まりはクルマに乗っていたときのことである。クルマってというのは、かの有名な(?)アルシオーネのこと。ここしばらく顔を見せないと思ったら、相変わらずクルマを乗りまわしている筆者なのでありました。

僕がクルマに乗るようになってからもう3年以上になる。この3年のうちに、ベイブリッジが開通したし、首都高速神奈川線の狩場線も(ローカルな話題でゴメン)開通した。道路自体は次第に整いつつあるように見える。

しかし、しかしだ。これはあくまで個人的な感想なのだけれども、渋滞は年々ひどくなっていったような気がする。確かに道路の整備によって交通事情が改善されたケースも、少ないながら、ある。完全とはいえないけれども、横浜新道がそのいい例だろう。

が、その一方で一般道路での渋滞は年々激化しつつある。特に交差点の近辺がひどい。最近目立ってひどいのが日比谷通りの芝公園付近だ。あ、またまたローカルな話題で申しわけない。でも編集室も芝公園とは近所の泉岳寺になったことですし、許してくださいませ。

交通事情が悪化してきているというのは最近いわれ始めたことではないのだが、個

人的には最近特にひどいような気がする。なぜなんだろう。直接的な原因としては車両の増加があるのだろう。なんたって最近のクルマの売れっぴりたるやすさまじいものがある。

でもそれと同時に、もうひとつの間接的な理由があるような気がする。表現が難しいのだけれど、「交通無策」とでもいおうか。要するに、交通をうまく流すための十分な対策が立てられていない、ということ。

具体的にいうと、車線規制、信号の可変制御なんかのことだ。確かにこれらの対策は立てられつつはあるのだけれど、十分でない、というところがほとんどなのだ。首都高速の箱崎なんか、特に有名だよ(ね(TVでもとりあげられているから知っている人も多いだろう))。

そろそろ本題に戻らないと、話がどこかに行ってしまう。要するに僕がクルマの中でなにを考えたのか。僕はこう思ったのだ。渋滞をコンピュータでシミュレートすることはできないのだろうか。そしてうまくいけば、渋滞解消法を見つけられないだろうか、と。

こうして僕は、コンピュータシミュレーションの世界への第一歩を踏み出した。

WHAT

まず僕が考えてみたのは、交差点の様子をうまくシミュレートできないだろうかということだ。これには理由がある。クルマに乗っていてよく通る交差点があるのだが、日中はいつも渋滞している。

よく観察した結果、これは信号のタイミングが悪いのではないかと考えるようになった。図1のような交差点なのだが、脇道βが合流してくるところで非常に混雑する。

素人考えて悪いのだが、自分なりに渋滞の仕組みを推測してみた。まず信号1が赤になって本線αの流れが止まる。逆に脇道β

からは図のAの部分にクルマが流れ込んでくる。ここで信号2も赤になると、図のAの部分にクルマが滞り、すぐにAはクルマで一杯になってしまう。

しばらくのちにまず信号1が青になる。が、BにいるクルマはAが一杯になっているために動けない。信号2も青になり、ようやくAにいるクルマが流れ始めてBが進もうとすると、ちょうど信号1が赤に変わってしまう。

このようにして、脇道からのクルマが比較的コンスタントに流れるのに対し、本線はほとんど動けない。その結果、激しい渋滞となる、というわけだ。

もちろん、この渋滞の第一の原因はクルマが多すぎることにあることは確かだ。その証拠に、クルマの台数が少なくなる夜間はこの交差点もスムーズに流れている。しかし、それは十分に理解したうえでも、この信号によっても渋滞が加速されているような気がしてならない。

所詮は素人のあさはかさなのかもしれない。実はこの信号の動作には、素人には理解しえない奥深い理由があるのかもしれない。確かに、知識だけからいえば、交通制御の専門家にはかなわないと思う。でも、我々には強い味方がいる。そう、コンピュータだ。この交差点の様子をコンピュータの中に再現できたら……。

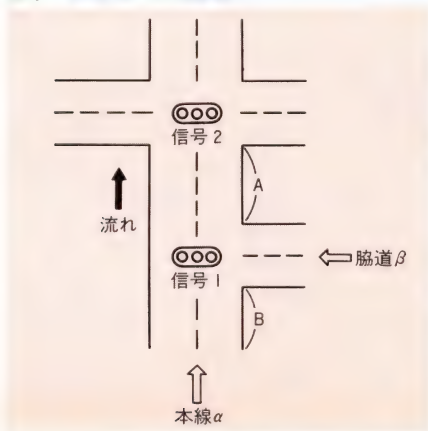
うまくシミュレートしてやれば、この素人考えを実証することもできる。さらに、シミュレーションの結果を利用して、信号をうまくコントロールしてやれば、あわよくば渋滞を緩和させることができるかもしれない。

HOW

さて、渋滞を解消させるなどと目標は高く掲げてみたものの、どうやってシミュレートしてやればよいのだろう。

まず最初に考えてみたのは、クルマ1台

図1 渋滞名所の交差点



1台の動きを精密にシミュレートできないだろうか、ということである。1台1台がシミュレートできれば、当然その集合体である交差点もシミュレートできるはずだ。

クルマの動きをよく見ているとわかると思うが(もちろんクルマを運転している人ならなおさらわかると思うが)、クルマの速度はだいたい先行車の速度と、先行車との車間距離によって決まる。え、先行車がない場合は、って。そんなときは先行車はかなり遠くを標準的な速度(60km/hぐらい)で走っていると考えてやればいい。もっとも渋滞シミュレーションじゃあ、なかなかそんな状況は生じないだろうけれど。

ま、とにかくこのように考えてみると、交差点に入ってくるクルマ1台1台に対し、先行車との関係(すなわち、先行車の速度と先行車との車間距離)から速度を割り出してシミュレートする、というモデルが考えられる。

このモデルの命となるのが、先行車との関係から自車の速度を割り出すという部分。クルマが走っている様子を頭に思い浮かべてみよう。前のクルマがどんどん離れていったらどうするか。まあ、普通は加速して追いつこうとするでしょう。

これは要するに、

IF 車間距離 = 増加 THEN 加速
のように考えることができる。同様にして、
IF 車間距離 = 一定 THEN 速度維持
IF 車間距離 = 減少 THEN 減速
であるかのように思える。

しかし、これでは間違い。なぜなら、車間距離がずいぶん開いてしまった場合には、「加速」して車間距離を「減少」させようとするだろう。

これはむしろ、先行車との適正な車間距離というものを設定し、実際の車間距離が適正值より大きかったら加速、などと考えたほうがよさそうである。

つまり、

IF 車間距離 > 適正值 THEN 加速
IF 車間距離 = 適正值 THEN 定速
IF 車間距離 < 適正值 THEN 減速
と考えることができる。

さて、これで骨組みはできた、さあプログラムを書こうというのは少し気が早い。まず車間距離の適正值はどうするか。これは当然車速に応じて変わってくるだろう。高速のときは通常、車間距離を多めに取るものだけれど、低速時は車間距離は少なめになってくるだろう。さらに完全に止まってしまった場合は、だいたい1mぐらいが適正值だろう。

ということは、適正值は車速の関数として書けることになる。すなわち、

適正值 = f (車速)

というわけ。

まだ問題はあ。加速するといっても、どのように加速するのだろうか。どれぐらい車間距離が変動しているのかによっても加速の度合いは変わってくるはずだ。具体的にいえば、車間距離が大幅に適正值を上回っていたら派手に加速するだろうし、少しだけだったら徐々に加速するだろう、ということ。

さらに加速の際には最初は加速度が大きく、徐々に加速度が小さくなっていく、ということも考えねばならない。

以上を総合してみると、加速の大きさは車間距離の関数になる。さらに加速の際の加速度の変動のしかたは、独自の関数に従うことになる。そしてその加速度を積分してやれば、求めたい速度が、ようやく、得られる(図2を参照してほしい)。

こうして見てみると、このシミュレーションは関数の嵐になることがわかる。まあシミュレーションなんていうのはもともと演算の積み重ねでできるものだから、当たり前といえば当たり前なのだけれども。

ところが、実際にプログラムを書いてみたところ、大パニックに陥ってしまった。とにかく複雑になってしまい処理しきれなかったのである。いろいろ簡素化して実現しようとしてみたのだが速度的にも辛いものがある。

下手をすると実際の交差点モデルよりも遅くなってしまいそうだったのである。もともとシミュレーションは将来をより簡単に、より早く知るために役立つはずのものなのであるから、これでは意味がない。

結局、あろうことか僕は匙を投げ出してしまったのである。しかしもちろん、努力次第ではこのやり方でうまくいくはずである(誰かやってみてください)。しかし、面倒臭い!

さて、こうしていきなり試みは挫折してしまったのである。この連載の運命やいかに……。

STEP BACK

これはなんについてでもいえることだけれども、行き詰まってしまった場合はどうすべきだろうか。

まあ、普通は一步下がってもう一度よく考え直してみる、といったところだろう。そうすることによって新しい方法が見えて

くこともある。例にもれず、ここでももう一度いろいろと考えてみることにする。

物事をシミュレートする場合、現実をどのように模倣するか、というのが大きなポイントになる。いい換えればどのようなモデルにするか、ということである。

ひと口にモデルといっても、2種類のモデルを考えることができる。ひとつは「物理的モデル」という代物である。この代表例が大きさを縮小するスケールモデルである。交差点のシミュレーションの例でいえば、10分の1の大きさのクルマや信号などを作って動かすことによりシミュレートするモデル、ということになる。

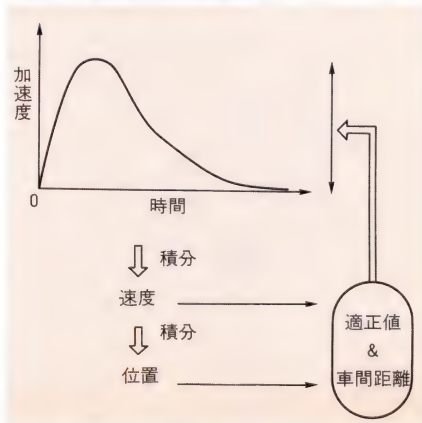
もうひとつは「数値モデル」である。これは現実を数値に置き換えることによって構成される抽象的なモデルのことで、「数値を扱うことならおまかせ」というコンピュータでシミュレーションをするとなると、当然数値モデルを考えることになる。

ただ、数値モデルといっても1種類とは限らない。現実の要素をどのようにとらえるかで異なるモデルができてくる。その要素のなかでもっとも重要なものが「時間」であり、その「時間」をどのようにとらえるかで2種類のモデルができあがる。

違いは時間を連続するものとするか、あるいはポイント、ポイントがつながったものとするのかという点にあり、前者は「連続変化モデル」、後者は「離散変化モデル」と呼ばれている。

連続変化モデルは、時間を連続的に(もちろんコンピュータ上では微小な等時間間隔で)変化する値であると考え、時間を中心にモデルを変化させていく。逆に離散変化モデルでは時間というものは、モデルが変化する重要なポイントをつなげたものとして表現される。いい換えればモデルの変化に応じて時間が(不等時間間隔で)刻まれていくということになる。

図2 加速度算出の概念図



あえて簡単にいってしまうと、「いま時間がこうだから、モデルはこうなる」というのが連続変化モデル、「モデルがこうなったから、時間はこうなる」というのが離散変

The diagram illustrates a 3-lane automated lottery machine layout. At the bottom, a 'クルマ' (Car) icon is shown with an upward arrow, labeled '3±2秒に1台' (1 machine every 3±2 seconds). The machine body contains three lanes, each with a '券機' (Ticket Machine) and two 'クルマ' (Car) icons. Lane 1 (right) is labeled '①', Lane 2 (middle) is labeled '②', and Lane 3 (left) is labeled '③'. Arrows point from the '券機' to the 'クルマ' icons. A label '左ハンドル専用' (Left-hand drive only) points to the leftmost 'クルマ' icon, and '右ハンドル専用' (Right-hand drive only) points to the rightmost 'クルマ' icon. A label '処理時間 7±2秒' (Processing time 7±2 seconds) points to the '券機' in Lane 1.

ここまでくればもうおわかりでしょう。
連続変化モデルに挫折してしまった以上、
ここは離散変化モデルにアタックするしか
ない！ わけだ。

ここでは、遠回りにはなるかもしれない

は通行券の発券のみ。発券用の車線は3車線あって、そのうち2つ（第1レーン、第2レーン）が右ハンドル専用、すなわち右側に発券機があり、残りのひとつ（第3レーン）が左ハンドル専用、すなわち左側に発券機が設置されている。

それではこのモデルをシミュレートしてみることにしよう。シミュレーションの前提として、発券にはどの発券機でも5~9秒（7±2秒）のあいだの時間（整数秒）を等確率でとり、クルマは混雑時らしく1~5秒（3±2秒）間隔でやってくるとする。なお、クルマ全体に占める左ハンドルの割合は10%であるとする。

ICに入ってくるクルマは左ハンドルなら無条件に第3レーンに進む。右ハンドルの場合、第1レーンか第2レーンのうち、すいているほうに進む。もし同じように混んでいる場合は等確率で第1レーンか第2レーンのどちらかに進む。

以上の条件に従ってBASICでプログラムを書いてみたのがリスト1である。今回は、どのようにしてプログラムを組んでいくかということよりも、シミュレーションがどのように有効かを中心に見ていこうと思う。そこでさっそく表1がリスト1の実行結果である。

おっとその前に新しい言葉を3つ覚えてもらおう。その3つとはトランザクション（Transaction, 略してXact）、ファシリティ（Facility）、キュー（Queue）のこと。順番に説明していこう。

トランザクションとは「時間の経過とともにシステムを動いていく対象」をモデル化したもののこと。というとし難いそうだが、シミュレーションのシステムの中を動いていくものだから、システムの「コマ」とでもいおうか。この例の場合はシミュレーションのシステム（IC全体）の中を移動していくもの、すなわち、クルマがトランザクションということになる。

システムの中を動いていくトランザクションに対し、ファシリティは動かない対象をモデル化したものである。定義は「同時に単一のトランザクションのみが使用する機器設備類」ということになる。

動いてくるトランザクションを処理していくのがファシリティということになるが、同時にひとつのトランザクションしか利用できない、というところがミソである。トランザクションをシステムの「コマ」にたとえたとすると、ファシリティは「マス」とでもいおうか。

この例でいえばシステム（IC）の中にあ

って、トランザクション（クルマ）が利用する対象であるから、「発券機」ということになる。発券機は、もちろん、移動しないし、同時に1台のクルマしか利用できない。

さて、発券機を利用するのはいいのだが、必ずしもすぐ利用できるとは限らない。ときには混雑しているため列に並んで待たなければならないこともあるだろう。このとき生ずる「待ち行列」のことを指してキューと呼ぶ。

以上をまとめると、「システムの中をトランザクションが移動していく。その途中でファシリティを使用するわけだが、ファシリティは同時にひとつのトランザクションしか使用できないから、待たなければ使用できないこともある。そのため待ち行列キューができる」ということになる。どうです、わかりました？

それでは以上を頭にしっかり刻みこんだうえで表1を見てみよう。

まずシミュレーション時間。ほぼ3000秒といったところだ。これはこのシミュレーションが1000個のトランザクション（1000台のクルマ）を処理するまで、ということで行われていることから理解できるだろう。平均3秒に1台クルマがやってくるのだから、3秒×1000台は3000秒ということだ。

さて次はファシリティに関する結果である。最初はファシリティの番号。1、2、3はそのまま図の車線番号に対応している。ということはファシリティ1、2は右ハンドル専用の発券機、ファシリティ3は左ハンドル専用の発券機ということになる。2項目めのアベレージュエティリゼーションというのは「平均使用効率」という意味である。すなわちシミュレーション実行中、（時間比で）どれぐらい使用されていたかということを表している。

これを見るとシミュレーション実行中（約3000秒）、ファシリティ1、2は99.5%も使用されていたことになる。逆にいえば、発券機が空いていたのは0.5%程度でしかない。

表1 リスト1(改良前)の実行結果

Result									
Simulation time (t) : 3092 s									
-- Facility --									
name	average		number	average					
or no.	utilization		entries	time / Xact					
1	0.9951		445	6.930				in use	
2	0.9942		440	7.002				in use	
3	0.2717		117	7.179					
-- Queue --									
no.	max	average	total	zero	percent	average	current		
	contents	contents	entries	entries	zero	time /Xact	contents		
1	17	7.283	460	5	0.011	48.952	15		
2	18	7.320	456	7	0.015	49.636	16		
3	2	0.037	117	82	0.701	0.966	0		

それに比べてファシリティ3は30%すらも利用されていない。ファシリティ1、2が混雑しているのをよそ目に、ファシリティ3はガラガラだったのである。

3項目めはシミュレーション終了までに何台のクルマがそれぞれのファシリティを使用したか、を表している。1、2は当然ほぼ同じで、合計で全体の9割ほど（=右ハンドルの割合）、3は左ハンドル専用であるから当然左ハンドルの割合1割にほぼ等しくなる。

ここで注意してほしいのだが、ファシリティ3の割合は設定の「左ハンドルは10%」には正確には合致しない。これはシミュレーションに乱数を用いているからである。考えてみればわかることだが、左ハンドルが10%だからといって、まず右ハンドルが900台きて、その次に左ハンドルが100台くるわけではない。お互いにごちゃまぜになっていて、全体として割合を見れば10%である、というふうになる。そこでコンピュータ上では乱数を用いて右/左ハンドルをうまく配分しているわけだが、乱数を用いている以上多少のゆらぎが出てきてしまうのである。

これは左ハンドルの割合だけの問題ではなく、結果すべてに共通することである。だから読者がリスト1を入力して実行しても表1とまったく同じ答えは出てこないだろう。傾向は同じでも多少の差は当然なのである（ましてこのような渋滞シミュレーションでは、渋滞が渋滞を呼んで大きな差を生ずることもある）。「あれ、ちゃんと入力したのに答えがあわない！」などとあわてないように。

さて、本題に戻るとしよう。次の項目はトランザクション1個（クルマ1台）を処理するのに要した時間の平均である。1、2、3ともほぼ7秒であるが、これは発券機の処理時間はどれも7±2秒である、という前提に合致している（正確に7秒にはならない理由は上で述べたとおり）。

さらに、最後に「in use」と表示されてい

るのは、そのファシリティが現在使用中であることを示している。

それでは次はキュー（待ち行列）の結果を見ていくことにしよう。まずは待ち行列番号であるが、これはそのままファシリティの番号に一致している。2項目めは最大どれぐらいの長さの待ち行列（現在使用中のトランザクションを除く、純粹に待っているトランザクションの数）ができたか、を示している。1、2はなんと最大で20台近くも並んだことになる。それに引きかえ3は最大でも2台しか待っていない。

3項目めは平均してどれぐらいの長さの待ち行列ができていたかである。前項は最大を表しているが、こちらは平均である。これによると1、2には大体いつも7、8台並んでいたことがわかる。一方3はといえば、ほとんど待ちなしだったことがわかる。

その次は待ち行列に入ったトランザクションの数である。前提から当然、約9対9対2の比率になっている。ファシリティのときの値より少し大きいのは、待っているうちにシミュレーションが終わってしまったクルマも含まれているから。

さて、次の2項目は待ち行列に入ったがファシリティが使用されていないので待ちなしで利用できたトランザクションの数とその割合である。容易に想像がつくように

1、2ではほぼすべてのクルマが待たなければならないのに対して、3は70%は待ちなしという有様である。

その次はもっとも重要な結果、すなわち平均待ち時間である。1、2は平均で50秒近くも待たなければならないのに、3は1秒も待たない。なんたる不公平！

そして、ようやく、といった感じの最後の項目だが、これはシミュレーションが終了した時点でどれだけの待ち行列ができていたか、を表している。かわいそうに1、2ではいまも15台ものクルマが待ち続けているのだ。

and...

さて、こうしてひととおりシミュレーションの結果を見てきた。どう感じただろうか。なにに左ハンドルだけが優遇されすぎている、って。ごもっとも、1割しかない左ハンドルに1レーンまるまる提供しているのだから、さもありなん。

確かに結果から明かなように、右/左ハンドルで平均待ち時間の格差が大きすぎる。これでは右ハンドルのドライバーが黙っていないだろう。

Ok、確かに君のいうとおり、ではどうするね。結果から問題点を指摘して終わるかい？

それもまたひとつの道だが、あまり賢明とはいえないだろう。ここで初心にたちかえって、なんのためにシミュレーションをするのかを考えてみよう。シミュレーションは目的があって行うもの。なにかのシステムのこんなところを改良したい。でも現実に調査して改善を図るのは効率が悪い。そんなときにこそシミュレーションで効率よく改善を図ろうとするのだ。

それではどうするか、もうおわかりだろうと思う。改善策を練り、シミュレーションで確かめてみよう。

上のシステムの欠陥は左ハンドルを優遇しすぎることにあった。これは第3レーンを左ハンドル専用にしてしまったからだ。それじゃあ第3レーンを左/右ハンドル兼用にしてみたらどうだろうか。第3レーンの右側にも発券機を設置すれば左右両側にあることになり、左ハンドルでも右ハンドルでも利用できる。

さて、図4が改良後の料金所である。プログラムはリスト1にリスト2をつぎはぎをするかたちで利用しよう。入力する際にはまずリスト1を入力し、改良に対応するためにリスト1のうち10070行から10140行までをリスト2のように書き換え、さらにリスト2の30000行以降をリスト1の最後に加えてほしい。はい、できあがり。

こうして得られたプログラムの実行結果

リスト1

```
1000 'Simulation model 2 ver.1.01
1010 '
1020 ' for X1 BASIC
1030 '
1040 ' 1990.10 (c) Cammon
2000 '
2010 ' initialize : set constant
2020 WIDTH 80: CLS
2030 DEFINT a-z
2040 t:=0: nxact:=0
2050 pgen1=3: pgen2=2
2060 ptran=1
2070 p2(1)=7: p2(2)=7: p3(1)=2: p3(2)=2
2080 p2(3)=7: p3(3)=2
2090 oxact=1000
3000 '
3010 ' main
3020 REPEAT
3030 LOCATE 0,0: PRINT USING " t : ####";t!
3040 IF t!=nxact! THEN GOSUB 10020
3050 FOR i=1 TO 3
3060 GOSUB 20020: IF dxactt=oxact THEN i=3
3070 NEXT
3080 LOCATE 0,1: PRINT USING "Xact : ####";dxactt
3090 t:=t!+1
3100 UNTIL oxact=dxactt
3110 t:=t!-1
3120 GOSUB 25020
3130 END
10000 '
10010 ' generate Xact
10020 nxact! = t!+pgen1-pgen2+INT(RND(1)*(2*pgen2+1))
10030 xact=1
10040 IF INT(RND(1)*10) < ptran THEN p1=3: RETURN
10050 GOSUB 10070: RETURN
10060 '
10070 p1=INT(RND(1)*2)+1
10080 IF u(1)=0 AND u(2)=0 THEN RETURN
10090 IF u(1)=0 AND u(2) THEN p1=1: RETURN
10100 IF u(1) AND u(2)=0 THEN p1=2: RETURN
10110 IF q(1) = q(2) THEN RETURN
10120 IF q(1) > q(2) THEN p1=2 ELSE p1=1
10130 RETURN
10140 '
20000 '
20010 ' queue
20020 quwtt!(i)=quwtt!(i)+q(i)
20030 IF xact AND p1=i THEN GOSUB 20110
20040 IF u(i)=0 OR t!<qfreet!(i) THEN RETURN
20050 xacttt!(i)=xacttt!(i)+qfree(i)
20060 IF q(i) THEN q(i)=q(i)-1: GOSUB 20150 ELSE u(i)=0
20070 dxact(i)=dxact(i)+1: dxactt=dxactt+1: RETURN
20080 '
20090 'This program is dedicated to Saeko.Yes,it's you!
20100 '
20110 xact=0: nque(i)=nque(i)+1
20120 IF u(i) THEN q(i)=q(i)+1: GOSUB 20180: RETURN
20130 u(i)=1: nqze(i)=nqze(i)+1: GOSUB 20150: RETURN
20140 '
20150 qfree(i)=p2(i)-p3(i)+INT(RND(1)*(2*p3(i)+1))
20160 qfreet!(i)=qfree(i)+t!
20170 nfce(i)=nfce(i)+1: RETURN
20180 IF q(i)>maxq(i) THEN maxq(i)=q(i)
20190 RETURN
25000 '
25010 ' print result
25020 PRINT "Result": PRINT
25030 PRINT "Simulation time (t) :";t!;"s": PRINT
25040 PRINT " -- Facility --"
25050 PRINT " name average number average"
25060 PRINT " or no. utilization entries time / Xact"
25070 FOR i=1 TO 3
25080 PRINT USING " #";i;
25090 PRINT USING " ###.###";xacttt!(i)/t!;
25100 PRINT USING " #####";nfce(i);
25110 PRINT USING " ###.###";xacttt!(i)/dxact(i);
25120 IF u(i) THEN PRINT " in use";
25130 PRINT
25140 NEXT
25150 PRINT: PRINT: PRINT " -- Queue --"
25160 PRINT " no. max average total zero perc"
25170 PRINT " ent average current contents contents entries entries zer"
25180 PRINT " o time /Xact contents"
25190 FOR i=1 TO 3
25200 PRINT USING " #";i;
25210 PRINT USING " ###.###";maxq(i);
25220 PRINT USING " #####";quwtt!(i)/t!;
25230 PRINT USING " #####";nque(i);nqze(i);
25240 PRINT USING " ###.###";nqze(i)/nque(i);
25250 PRINT USING " ###.###";quwtt!(i)/nque(i);
25260 PRINT USING " ###";q(i)
25270 NEXT
25280 RETURN
```


が表2である。システム3と4で実行条件は基本的に同じである。違いは4では第3レーンが左右兼用になっているため、第3レーンがすいていれば右ハンドルのクルマもどんどん利用する、という点にある。

それでは実行結果を見てみよう。ひと見ればわかると思うが、状況は劇的なまでに改善されている。平均待ち時間はどのレーンでも約2秒。改良前は50秒近かったのとは比べると天と地ほどの差がある。これだけわずかな改良でこれほど効果があるとは正直いって筆者も予想していなかった。

ただ、今回は逆に第3レーンのほうが使用効率も高く、待ち時間も（わずかながら）長くなっている。これは第3レーンは右ハンドルに加え、1割いる左ハンドルも処理しなければならないからである。すなわち今度は逆に、左ハンドルのほうが待ち時間が長くなってしまったのである。

しかし左ハンドルの待ち時間が長くなったといっても微々たるものだし、多数派の右ハンドルの状況は大幅に改善されている。どちらのモデルのほうがよいかは明白だろう。当然後者だし、実際の料金所も後者のようになっている（はず）だ。

* * *

以上、シミュレーションの実例を見てきたわけだが、今回はどのようにシミュレーションを実現するか、というよりも、シミュレーションがどのように有効なのかを中心に考えてきた。シミュレーションをうまく使ってやれば、システムの改善の大きな力になることがわかってもらえたと思う。

ただ、こんな漠然としたモデルで、本当に現実を反映しているのだろうか、と考えている人もいるだろう。確かに今回のモデルは多分に漠然としていて、条件もあまり厳密なものとはいえない。でもこんなモデルでも、ちょっとした改良で劇的に状況が改善されることぐらいはわかる。まあ、これぐらいのモデルなら傾向がつかめれば十分だろう。

もし、もっと厳密な結果がほしいのであれば、条件をさらに厳密にしてやればよいだけのこと。その際でも原則はまったく同じである。

NEXT

今回は導入部ということで、シミュレーションの概略を見てきた。次回はどのようにシミュレーションを実現させるのか、から始めて、もっと複雑な例にも挑戦してみたいと思う。では、また。

図4 料金所の様子（改良後）

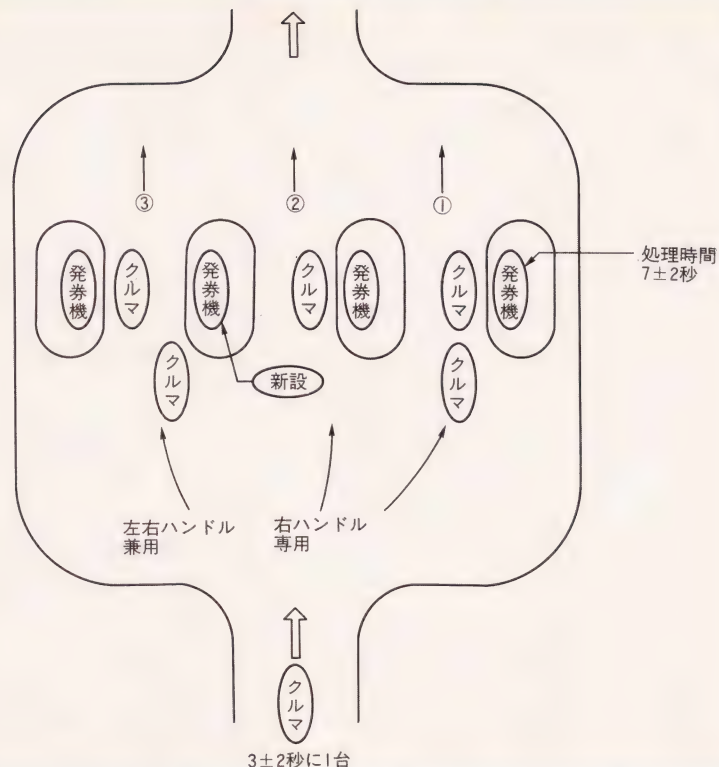


表2 リスト2(改良後) の実行結果

Result

Simulation time (t) : 2945 s

-- Facility --

name or no.	average utilization	number entries	average time / Xact	
1	0.7725	326	6.979	
2	0.7766	327	7.015	in use
3	0.8411	349	7.118	in use

-- Queue --

no.	max contents	average contents	total entries	zero entries	percent zero	average time / Xact	current contents
1	3	0.191	326	196	0.601	1.724	0
2	3	0.181	327	195	0.596	1.627	0
3	3	0.332	350	155	0.443	2.794	1

リスト2

```
10070 m(1)=u(1): m(2)=u(2): m(3)=u(3): GOSUB 30020
10080 IF flg THEN p1=min: RETURN
10090 IF min THEN ON min GOSUB 35010,35020,35030: RETURN
10100 IF u(1)=0 THEN GOSUB 35040: RETURN
10110 m(1)=q(1): m(2)=q(2): m(3)=q(3): GOSUB 30020
10120 IF flg THEN p1=min: RETURN
10130 IF min THEN ON min GOSUB 35010,35020,35030: RETURN
10140 GOSUB 35040: RETURN
```

```
30000 '
30010 ' minimum routine
30020 IF m(1)<m(2) AND m(1)<m(3) THEN min=1: flg=1: RETURN
30030 IF m(1)<m(2) AND m(1)>m(3) THEN min=3: flg=1: RETURN
30040 IF m(1)<m(2) AND m(1)=m(3) THEN min=2: flg=0: RETURN
30050 IF m(2)<m(3) AND m(2)<m(1) THEN min=2: flg=1: RETURN
30060 IF m(2)<m(3) AND m(2)>m(1) THEN min=1: flg=1: RETURN
30070 IF m(2)<m(3) AND m(2)=m(1) THEN min=3: flg=0: RETURN
30080 IF m(3)<m(1) AND m(3)<m(2) THEN min=3: flg=1: RETURN
30090 IF m(3)<m(1) AND m(3)>m(2) THEN min=2: flg=1: RETURN
30100 IF m(3)<m(1) AND m(3)=m(2) THEN min=1: flg=0: RETURN
30110 min=0: flg=0: RETURN
35000 '
35010 p1=INT(RND(1)*2)+2: RETURN 'generate 2 or 3
35020 p1=(INT(RND(1)*2))*2+1: RETURN 'generate 1 or 3
35030 p1=INT(RND(1)*2)+1: RETURN 'generate 1 or 2
35040 p1=INT(RND(1)*3)+1: RETURN 'generate 1 or 2 or 3
```


テクニックは偉大なめだ!

Komura Satoshi 古村 聡

とりあえず「3Dダンジョンを描く」ということにテーマも決まり、今月からいよいよぱていハンズの第2部が始まります。今後ともよろしく。ショートプログラムはX68000の画面クリアツールとX1のアクションゲームです。

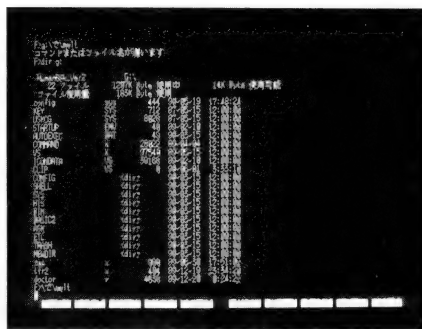


illustration : T. Takahashi

いやあ、先月はお休みになってしまったわけですが、別に連載が打ち切りになってしまったのではないのでご安心を。まあ、ほかのページにはちょこちょこ登場していたのでファンの皆さんもさびしくはなかったでしょう（ファンなんかいないって？グッスン）。

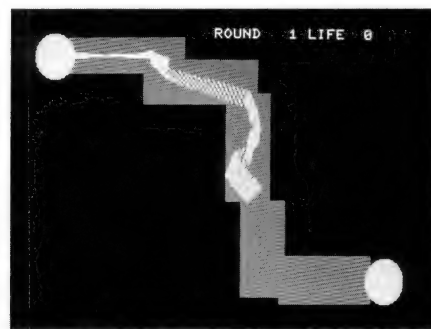
さて、季節はすっかり秋深し、という感じですよー。みんなどうやって時間つぶしているんでしょうか。私の場合はやっぱりプログラミングとかパソコンに向かいっぱなしということなんですけど……。少しわびしい。

しかし、それだけというわけじゃない。



とろける.X

私には原稿書きという天から与えられた使命がある（そんなものを与えた覚えはないとかいわれそうだけど）。特に、この2,3カ月というものはハンズのネタに苦しんでいた



KURUPER

んです。まあ、それもハンズを読めばわかるようにある読者の方によってなんとかなりました。つくづく、他人任せな（で）だなあと自分のことながら思ったりするので

リスト1 とろける.X

```
1: * テキスト画面が溶ける Ver 1.01
2: * MELT
3: * 1990/08/15
4: * By T.A.K
5:
6: .include doscall.mac
7: .include iocscall.mac
8:
9: .text
10: .even
11: inimlt:
12: lea.l w_fnt,a1
13: start:
14: dc.w $fe0e *乱数発生 d0
15: * $FEOE -- RAND
16: * 返り値 乱数（ロングワード符号付き）
17: * このコールはfloat2 or 3が必要です。
18:
19: move.l d0,d3 *乱数をD3へ
20: and.l #$3ff,d3 *D3を0~767に
21: and.w #$100,d0
22: beq go
23: and.w #$1ff,d3
24: go:
25: move.l #1,d1 *テキスト設定
26: IOCS _TCOLOR
27: move.l #$e0ff80,vraddr *set vram addr
28: move.l d3,d1 *X座標設定
29:
30: bsr zurasi1 *プレーン1
31: bsr zurasi2
32: bsr zurasi1
33:
34: moveq.l #2,d1
35: IOCS _TCOLOR
36: move.l #$e2ff80,vraddr *
37: move.l d3,d1
38:
39: bsr zurasi1
```

```
40:
41: bsr zurasi2
42: bsr zurasi1
43:
44: IOCS _B_KEYSNS *押されたか?
45: tst.w d0
46: beq start
47: move.w #16,d1
48: move.w #2,-(sp)
49: move.w #10,-(sp)
50: DOS _CONCTRL *画面をきれいに
51: move.w #0,-(sp)
52: move.w #14,-(sp)
53: DOS _CONCTRL
54: DOS _EXIT *終わり
55:
56: *ブロック転送
57: *in (a2)
58: *break (a2)+
59: zurasi2:
60: clr.w d2
61: IOCS _TEXTGET
62: move.w #2,d2
63: bra puttx
64:
65: zurasi1:
66: clr.w d2
67: IOCS _TEXTGET
68: move.w #1,d2
69: IOCS _TEXTPUT
70: addq.w #1,d1
71: rts
72:
73: vraddr:
74: .dc.l $e00000
75: w_fnt:
76: .dc.w 1,511
77: .ds.w 64
78:
79: .end
```


あります。

さて、たわいもない話はこれぐらいにし
て(?)本題のほうをぼちぼちいきますか。



画面がドねりん

ではでは、今月の1本目。まずはX68000
用の画面消去ユーティリティ「とろける。
X」です。そう、字が溶けるんです。

とろける。X for X68000

(要アセンブラ, リンカ)

広島県 一岡孝浩

このプログラムはアセンブラで書かれて
いますので、実行するためにはアセンブラ、
リンカ、それとDOSコール、IOCSコール用
のマクロが必要になります。アセンブラ、
リンカは福袋や、C compiler PRO-68K
Ver.1, Ver.2 などに入っています。ただ

しマクロは (iocscall.mac, doscall.mac)
は福袋にはついてきませんので、C com
piler PRO-68KあるいはOh!X 6月号の付
録のディスクに入っているものを使用して
ください。

プログラムを実行するには、まず、

A>ed とろける.S

としてエディタを立ち上げます。そしてリ
スト1を打ち込んでいってください。あ、
行番号は入れちゃだめですよ。打ち終わっ
たら、

[ESC]・E (ESCキーを押してからE
を押す)

としてセーブしてから、エディタを終了し
てください。

続いて、アセンブル&リンクをします。

A>as とろける

A>lk とろける

リスト2 とろける.Xの変更点

```
56: *ブロック転送
57: *in (a2)
58: zurasi2:
59:         clr.l    a1
60:         IOCS     _B_SUPER
61:         move.l   d0,a6
62:         bsr      xyaddr
63:         move.w   #102,d2
64:         move.l   #-256,a4
65: loop51:  bsr      loop512
66:         subq.w   #1,d2
67:         bne      loop51
68:         addq.w   #1,d1      *x++;
69:         move.l   a6,a1
70:         IOCS     _B_SUPER
71:         rts
72:
73: zurasi1:
74:         clr.l    a1
75:         IOCS     _B_SUPER
76:         move.l   d0,a6
77:         bsr      xyaddr
78:         move.w   #102,d2
79:         move.l   #-128,a4
80:
81: loop5:   bsr      loop512
82:         subq.w   #1,d2
83:         bne      loop5
84:         addq.w   #1,d1      *x++;
85:         move.l   a6,a1
86:         IOCS     _B_SUPER
87:         rts
88:
89: loop512: move.b   0(a3,a4),d4
90:         and.b   d0,d4
91:         beq     prese2
92: pset2:   or.b    d0,(a3)
93:         bra     decad2
94: prese2:   eor.b   $fff,d0
95:         and.b   d0,(a3)
96:         eor.b   $fff,d0
97: decad2:  lea      -$80(a3),a3
98:
99:         move.b   0(a3,a4),d4
100:        and.b   d0,d4
101:        beq     prese21
102:        or.b    d0,(a3)
103:        bra     decad21
104: prese21:  eor.b   $fff,d0
105:        and.b   d0,(a3)
106:        eor.b   $fff,d0
107: decad21: lea      -$80(a3),a3
108:
109:        move.b   0(a3,a4),d4
110:        and.b   d0,d4
111:        beq     prese22
112:        or.b    d0,(a3)
113:        bra     decad22
```

```
114: prese22:  eor.b   $fff,d0
115:        and.b   d0,(a3)
116:        eor.b   $fff,d0
117: decad22:  lea      -$80(a3),a3
118:
119:        move.b   0(a3,a4),d4
120:        and.b   d0,d4
121:        beq     prese23
122:        or.b    d0,(a3)
123:        bra     decad23
124: prese23:  eor.b   $fff,d0
125:        and.b   d0,(a3)
126:        eor.b   $fff,d0
127: decad23:  lea      -$80(a3),a3
128:
129:        move.b   0(a3,a4),d4
130:        and.b   d0,d4
131:        beq     prese24
132:        or.b    d0,(a3)
133:        bra     decad24
134: prese24:  eor.b   $fff,d0
135:        and.b   d0,(a3)
136:        eor.b   $fff,d0
137: decad24:  lea      -$80(a3),a3
138:
139:        rts
140:
141: *V R A M   アドレスの計算
142: *inp d1=X座標
143: *out (a3)=V R A M   アドレス
144: *      d0=マスクパターン
145: xyaddr:
146:        move.l   d1,d0
147:        move.l   d0,d5
148:        lsr.l    #3,d0
149:        add.l    vraddr,d0
150:        move.l   d0,a3
151:        and.l    #%111,d5
152:        lea.l    mskpat,a0
153:        move.b   0(a0,d5.1),d0
154:        rts
155:
156: mskpat:  .dc.b   %10000000
157:        .dc.b   %01000000
158:        .dc.b   %00100000
159:        .dc.b   %00010000
160:        .dc.b   %00001000
161:        .dc.b   %00000100
162:        .dc.b   %00000010
163:        .dc.b   %00000001
164:
165: vraddr:  .dc.l   $e00000
166: w_fnt:   .dc.w   1,511
167:        .ds.w   64
168:
169:        .end
170:
```

アセンブル、およびリンクが終わりまし
た。エラーやウォーニングは出ていません
ね? 出ていたら打ち間違いがあると思わ
れますから、間違いを見つけて直してくだ
さい。あ、あと、.macファイルがないとエラ
ーになりますから、ちゃんとプログラムと
同じディレクトリに置いてください。

これですべてよし、と。あ、そうだ実行
させる前に効果がわかりやすいように画面
に文字を出しておきましょう。まあ、

A>DIR

とかしてディレクトリを表示しておけばい
いかな。はい、それでは、

A>とろける

で実行させてみましょう。……おお、溶け
てゆく。適当なところでスペースキーを押
すと画面がクリアされます。溶けるって
えば、PDSにもMELT.X(tarazo lab氏作)



とかがありますよね。こっちのほうが不気味だけど。



アセンブラはいいぞ！

作者の一岡さんはアセンブラでプログラムを作ったのはこれが初めてなんだそうです。拍手拍手。投稿原稿によると“本当は直接テキストVRAMをいじりたかったのですが……”ということです。なるほど、これはIOCSを使っているのか。それじゃあってことで私、早速やってみました。リスト2がI/O直接叩くバージョンへの変更点です。リスト1の55行から69行までをリスト2と入れ替えてみてください。

えっと、なにをしているかというオリ

ジナルではIOCSをお願いして、ランダムに選んだX座標から、

X+0 縦に1ドットずり下げる

X+1 同じく2ドット下げる

X+2 もう1回1ドット下げる

ということを繰り返して垂れているように見せているわけです。こいつをスーパーバイザモード（これが直接I/Oをいじるモードね）で自分でしこしこ1ドットずつずり下げるようにしたんです。結構スピードも速くなったでしょ（期待してたよりは遅かったんだけど。まだまだ精進が足りないかな？）。

いやあ、やっぱりこれだよな、これ。マシン語の醍醐味はI/Oに始まりI/Oに終わるのです。自分のための自分のパソコン、やっぱり骨の髄までしゃぶってしまいたい。機能の隅から隅までズズ、ズイーっと使い切ろうとするにはやっぱり小回りの効くマシン語がベストなんですよ。

やっぱりマシン語は速い！ それにちょっと実行の順序を変えただけでも露骨にスピードが変わってきたりして一番面白い言語なんです、実は（と私は思う）。だから、みんな“スピードアップのテクニック”なんてのを極めてやろうとあがいたりしているわけだ、これが。

そういう世界であるからして、当然のように金言格言、先人の知恵というのも多数存在していて、いまでも「ループ展開は勝

利！」「案ずるよりイミディエイトは易し」「SUPERで万全」……。ばかもの！ 偏ったことを教えるんじゃない！ どうせなら「載ってるものはROMでも使え」とか……。いや、それも偏ってるんじゃない。うーん、ひとりでなにやってんだらう。それにマシン語がわからんとちっとも面白くないだらうな。

ちなみに、リスト2も結構偏ってます。さすがに512回ループ展開&場合分け4回なんていう暴挙には出なかったけどね。



はみ出ちゃダメよ

なにがなんだかわからなくなったところで（初心者を陥れる入門講座なんて、そうそうないぞ）、今月の2本目いきます。X1用のアクション(?)ゲーム、KURUPER（クルッパー）です。

KURUPER for X1シリーズ

(CZ-8FB01)

神奈川県 森貴之

新旧どちらのBASICでもOKです。画面上に■（ボックスフィルといえいいのだろうか?）が並んで道ができています。スタート地点をくるくる回っているのが主人公のクルッパー君です（ただのラインにしか見えない? いいの、ただのラインだもん）。くるくる回転してるからクルッパーというわけ。このクルッパー君を道からはみ

リスト3 KURUPER

```
10 CLS4:INIT:WIDTH 40:CLICK OFF:LI=2:LEV=.02
20 LOCATE12,12:PRINT"K U R U P E R !":AS=INKEY$(0)
30 IF AS="" THEN 20 ELSE M=VAL(AS):IF M<1 OR M>5 THEN M=1
40 CLS4:O=100:ZX=0:ZY=0:GOSUB310
50 ON M RESTORE 330,350,390,430,470
60 READ A:FORI=1TOA STEP 4:READ A1$,A2$,A3$,A4$
70 A1=VAL("&H"+A1$):A2=VAL("&H"+A2$)
80 A3=VAL("&H"+A3$):A4=VAL("&H"+A4$)
90 LINE(A1+32,A2)-(A3+32,A4),PSET,1,BF:NEXT I
100 CIRCLE(20,20),16,6:PAINT(20,20),6
110 CIRCLE(300,180),16,6:PAINT(300,180),6:X=20:Y=20
120 O=O+LEV:S=STICK(0)+STICK(1)
130 IF STRIG(0)+STRIG(1) THEN ZX=0:ZY=0
140 ZX=ZX+(S=4)-(S=6):ZY=ZY+(S=8)-(S=2)
150 C=6:GOSUB 230:X=X+ZX:Y=Y+ZY
160 X1=COS(O):Y1=SIN(O)
170 IF POINT(X-X1*8,Y-Y1*8)=0 THEN 240
180 IF POINT(X+X1*8,Y+Y1*8)=0 THEN 240
190 IF POINT(X,Y)=0 THEN GOSUB 240
200 C=2:GOSUB 230
210 IF X>295 AND Y>175 THEN 290
220 GOTO 120
230 LINE(X+X1*8,Y+Y1*8)-(X-X1*8,Y-Y1*8),PSET,C:RETURN
240 C=2:GOSUB 230:
250 IF LI=0 THEN 270
260 LI=LI-1:LOCATE14,12:PRINT"MISS !":PLAY"O4C3":PAUSE5:
GOTO 40
270 LOCATE15,12:PRINT"GAME OVER":PLAY"04C3DECEDDDDCC"
280 IF STRIG(0)+STRIG(1) THEN RUN ELSE 280
290 LOCATE17,12:PRINT"CLEAR":LI=LI+1:M=M+1:IF M>5 THEN M=
```

```
1 LEV=LEV+.02
300 PLAY"O4E2DCEDDDC":PAUSE10:GOTO 40
310 COLOR7:LOCATE18,0:PRINT"ROUND ";M:"LIFE "LI:RETURN
320 '--- ROUND 1 ---
330 DATA 20, 00,07,5F,20, 3E,17,9D,36, 83,2F,A8,7B, 90,7B
,B5,BF, B1,A2,FF,C4
340 '--- ROUND 2 ---
350 DATA 60, 00,0A,13,1C, 14,0C,AE,1A, AF,0A,C8,2B, C8,18
,E1,40, 69,34,D5,45
360 DATA 1A,40,86,51, 04,49,24,74, 1C,5F,3C,93, 35,7A
,7E,9E, 6D,6F,B6,83
370 DATA 98,7E,E1,92, BC,91,E1,A5, A3,A3,C8,B7, C4,AE
,E9,C2, EA,AA,FF,BE
380 '--- ROUND 3 ---
390 DATA 44, 00,0B,26,1E, 12,1E,41,31, 38,0A,67,1D, 55,1E
,84,31, 74,32,A3,45
400 DATA 95,46,C4,59, 80,5A,AF,6D, 60,6E,8F,81, 7D,82
,AC,95, 9B,96,CA,A9
410 DATA B4,A9,FF,BE
420 '--- ROUND 4 ---
430 DATA 44, 00,07,1A,22, E6,07,FE,22, E6,A8,FE,C3, 00,A8
,18,C3, 00,57,18,72
440 DATA E7,57,FE,72, 1A,10,E6,19, 19,60,E6,69, 19,B2
,E6,BB, EB,1A,FB,5E
450 DATA 04,6B,14,AF
460 '--- ROUND 5 ---
470 DATA 40, 04,10,54,18, 47,15,97,25, 89,22,DB,2A, B2,B0
,FB,B8, 73,A3,BC,B3
480 DATA 33,9E,7C,A6, 21,59,33,A6, DC,22,EE,6F, 21,59
,88,63, 75,64,DC,6F
```


出ないように2, 4, 6, 8のキーでゴールまで運んであげてちょうだいな。クルッパ君は細長いけど、くるくる回転してるからタイミングがすべて。広いところで待っていて、角度がよくなったら一気に道を渡っちゃえ!

そうそう、ひとついい忘れてた。クルッパ君には慣性が働いているので、あんまり勢いよく渡ろうとすると止まれなくなるので注意して……。ああ、遅かったか。面クリアで1UP, また、タイトル画面で1~5のキーを押すと面セレクトができます(つまり全5面あるのだ)。

慣性か、そういや最近慣性を使ったものが多いなあ。さすがにパズルとかは別みたいだけど、慣性のかかるパズルってあったかなあ。でも、いったいどんなものになるんだろ。

ところでこのゲームで遊んでいたら、「これショート? 見た目にきれいだね」「写真うつりよさそうな画面ですね」とスタッフ

の間でもなかなかの評判でした。うーん、テクニシャン。ラインの魔術師と呼んであげよう。ラインがくるくる回転しながら動いて、しかも残像みたいに軌跡が残っていて、しかもヒラヒラ舞っているように見えるんだよね。本当にいいので、ぜひとも打ち込んでみてください。

画面センスがいいっていうのも大切な要素ですよ。短いプログラムで画面をきれいに見せるのは難しいですけどね。センスを磨くのもテクニックのひとつかな。



(で)からのお願い

ううう。実は「とろける. X」の一岡さん、プログラムをディスクじゃなくてプリントアウトで送ってきてくれたのです。薄い封筒に(で)様なんて書いてあるからファンレターかと思ったじゃないですか(来るわけないか)。

あのね……、別に打ち込むのが面倒臭い

とはいわないけど(そんなこといったら読者の皆さんに申し訳ないもんね)、私が打ち込むんだよ。タイプミスしてエンバグ(デバックの逆で人のプログラムにバグを付け加えてしまうこと)しちゃうかもしれないんだよ。それが全国何十万、何百万部(かなりオーバーだな)も売れていくんだよ。恐ろしいでしょう?

だから、皆さんディスクで送ってきてください。ディスクの容量が余ってもったいないっていうんだったら、いっぱいPCMデータ入れてもかまわないし(あんまり意味がない?), 制作日誌などを入れてもいいんですから。

しかし、本当にエンバグしてないかな。なかなか不安な今月の(で)なのでありました。どっかに打ち間違いがあったらこそり教えてね。まあ、ほとんど(絶対?)大丈夫だと思いますがね。そんなこんなでまた来月。

精進せえよ。

(で)のぱーていハンズ第2部——(その1)

はい、皆さんお待たせしました(だれも待ってないって!? まあ、そういわずにお茶でもどうぞ)。ぱーていハンズ第2部のスタートです。

うう、やっと来ましたリクエスト。うれしいじゃありませんか。愛知県の白井達広さん、ありがとね。

では、さっそく読ませていただきます。なにに、

「1周年特別企画—どんちゃん騒ぎの部屋」にてリクエストよろしくと書かれているのに気づいたので考えました。

- 1) ウィ〇ード〇イのようなラインで構成された(壁でもいいけど)の3D迷路のRPGを作る(モンスターは出なくてもいい)
- 2) コマンド選択式のテキストアドベンチャーを作る
- 3) マップを配列変数に詰め込んだスクロールするRPGのようなものを作る

ふむ、どれも面白そうですね。1番はあれかな。要するにダンジョンとか3D迷路を描くプログラムを作って、あとはイロイロと付け足せばいいのかな。これがいいか。せっかく1番目に書いてくれたことだし。

てなわけで第2部は「ダンジョンを描くのだ」というセンでいきましょう。ネタが決まって、めでたし、めでたし。

方針なのである

まずはなにをどういうふうにつけていくか決めずばなるまい。作るのは3Dのダンジョン。これを配列上のデータにしたがって、画面上に描いていくとかすればいいのかな。で、マップデータなんだけどこいつはいろいろな種類を作って、ラインの色を変えたりメッセージを出せるようにしておこう。そうすれば将来的にはいろんなイベントも入れられるし、階段を作ったりもできるでしょ。今回は1階分のデータだけでほかの階や実際のイベントは各自で作っていただきます。あ、モンスターは出なくていいってことなんでモンスターもなしね(こらこら)。

よしよし、だいたいの方針は立った。ところでなにから作るのか? こういう場合は一番簡単そうで、しかも目に見えるところから手をつけるのが得策だったりするのです。なんでかっていうと、難しくって目に見えないところからやっちゃうと、バグが出た場合に修正が鬼ムズになっちゃうからなのです。

だって、実行結果が目に見えないんですよ。プログラムのどこが間違ってるのかプログラムリストとにらめっこ(まあ、どんなプログラム

のデバックも最終的にはこれをやるんだけどね)しなくちゃバグが絶対わからない。おまけに打ち間違いくらいはバグならいいけどハナからアルゴリズムを間違えてたなんてシャレにもならない事態になったらそれこそ悲惨なものね(まあ、簡単どころでもアルゴリズムが間違っているとシャレにならんかもしれないけど)。

さて、3Dダンジョンを描くにあたって、ダンジョンを描くステップを1つひとつ順に書いていくと、

自分のいる位置を確認する



自分のいる位置からどのくらいのところになが見えるのかを調べる



自分の位置から見えるところにあるものを描く

となるわけだな。こいつを、

- 1) 壁を描くルーチン
 - 2) 自分から見える壁を選んで1)に壁を描かせるルーチン
 - 3) 自分の位置から向いている方向を調べ、その方向にある一定の位置の壁が見えるものとみなして2)に壁を描かせるルーチン
- という順番で作っていくと、実際に壁を描く様子を少しずつ見ながらデバックができるのだな。

おお、これは目に見える部分から作っていく法則にしたがっているではないか。

こいつはボトムアップ法という名前がよく知られているプログラムの組み方だったりするので(ボトム、下位のルーチンから徐々に上のルーチンに上がる、アップしていくわけだな)、実はえらくもなんともなかったりするのだが……。

おお、重要なことを忘れていた。今回使用するのはX68000、言語はX-BASICなのであった。やっぱりなんだかんだいってもBASICが一番デバッグしやすいということ、初心者だろうがなんだろうがX68000ユーザーが全員持っているからという実にリーズナブルな選択なのである。一応、初心者向けということなので出来もスッポンと手抜きであることだし、中級者以上の方はアルゴリズムの説明だけ読んで自分できっちりとプログラムを組み、コンパイルするなりCに移植するなりしていただきたい。とりあえずそういうこと。

さっそく説明開始

では、実際のプログラムに入ります(最初だからといってプログラムに入らないほど私は甘くないのだ。まるで学校の意地悪な先生だやることが)。まずは目に見えるものということだから、ダンジョンのパターンを全部描き出してしまします。

最初に考えなくてはならないのが「何歩先まで見えることにするか」ということ。つまりバリバリ全開に視界が開けててもある程度先以上は見えなくしてしまうというわけなのね、これが。本当は無限の彼方に壁があったら地平線になるようにできればいいんだけどそこまでやってもあまり意味はない、ということである程度以上は見えないということにしようわけ。で、ここでは正面が4段階、横の壁は3段階まで見えるということにします。これをマップの状況に応じて組み合わせて壁を描いていくわけね(うーん、1対1対応。グラフ理論だな)。

あ、あと絵を描く前にすることがあった。画面の初期設定ってやつです。うんと、これはなにかというと、これからグラフィックで絵を

描くわけですから、“これから絵を描くぞー”という宣言をしてやらないかんわけです。

それにですね。画面モードというやつがありまして、なにを使いたいかによっていろいろ設定しなくちゃいかなのですね。で、とりあえず、画面に絵を描くぞー、画面は512×512ドットで16色使えるモードにするぞー、ということを宣言するにはSCREEN文というのを使います。SCREEN文に関してはBASICのマニュアルを参照のことね(SCREENに限らずわからない命令があったらマニュアルを見ようね)。

はい、おまたせ。それではみんな壁のパターンのお絵描きをしてみましょう。まず、画面一杯にボックスするようなパターンを作ります。それから少しずつ小さくなるように描いていくんですね。小さくなるパターンをあつ3つ、ボックスで描いてやります。そうそう、要するに遠近法ってやつです(あ、そうだ。本当の遠近法では遠くにいくほど壁の間隔は狭くなるんですが……。はっ、みんな等間隔になってる。ま、それでもちゃんと見えるからいいよね)。

次にサイドのパターンをさっき作った□にあわせて斜めの線をひとつずつ引いていきます。図1のような感じですね。右側全部描けました? それじゃ、壁が途中で切れているパターンね。横に壁がないところは図2のように横線を引きます。できたかな。あとは左にも同じようにするだけです。座標に気をつけてね。

おっと、忘れるとこだった。実際にはメッセージウィンドウのために画面下と右側に少しスペースを空けている。だから、正確には“ダンジョンの絵のエリア一杯のボックス”なのだ。よろしい?

というわけで壁のパターンができた。正面4パターン、サイド3×2(左右)パターン、サイドの切れ目3×2パターン。全部で16個の壁を描いたわけだ。ふーっ、さすがに私もつかれた。リストも結構長いし。

では、今月はこのへんにしておきましょう。おなかもへったことだし。来月はこれを使って目に見える範囲の壁を描くルーチンを作ります。とりあえず、

drawbox()

とか、

drawsl2()

とかやってみてどんな絵が描けるか試してみてください。

では、また来月このOh!Xで。ぱっくん(とあんぱんを食べてくる)。

リスト

```
280 /*その他の初期化*/
290 clrscr()
300 end
1070 /*画面の初期化*/
1080 func clrscr()
1090 screen 1,1,1,1
1100 box(0,384,511,511,15)
1110 locate 0,24
1120 print"運命の迷宮へようこそ"
1130 endfunc
2550 func drawbox1(wcol)
2560 box(0,0,383,383,wcol) /*深さ1*/
2570 endfunc
2580 func drawbox2(wcol)
2590 box(48,48,335,335,wcol)/* 2 */
2600 endfunc
2610 func drawbox3(wcol)
2620 box(96,96,287,287,wcol)/* 3 */
2630 endfunc
2640 func drawbox4(wcol)
2650 box(144,144,240,240,wcol)/* 4 */
2660 endfunc
2670 func drawsl1(wcol)
2680 /*左深さ1*/
2690 line(0,0,47,48,wcol)
2700 line(47,48,47,335,wcol)
2710 line(48,335,0,383,wcol)
2720 /*line(0,383,0,0,wcol)*/
2730 endfunc
2740 func drawsl2(wcol)
2750 /*左 2*/
2760 line(48,48,95,96,wcol)
2770 line(95,96,95,287,wcol)
2780 line(95,287,48,335,wcol)
2790 line(48,335,48,48,wcol)
2800 endfunc
2810 func drawsl3(wcol)
2820 /*左 3*/
2830 line(96,96,143,144,wcol)
2840 line(143,144,143,240,wcol)
2850 line(143,240,96,287,wcol)
2860 line(96,287,96,96,wcol)
2870 endfunc
2880 func drawsl1(wcol)
2890 /*右 1*/
2900 line(383,383,336,335,wcol)
2910 line(336,48,383,0,wcol)
2920 line(383,0,383,383,wcol)
2930 line(336,336,336,48,wcol)
2940 endfunc
2950 func drawsl2(wcol)
2960 /*右 2*/
2970 line(335,335,288,287,wcol)
2980 line(288,287,288,96,wcol)
2990 line(288,96,335,48,wcol)
3000 line(335,48,335,335,wcol)
3010 endfunc
3020 func drawsl3(wcol)
3030 /*右 3*/
3040 line(287,287,241,240,wcol)
3050 line(241,240,241,144,wcol)
3060 line(241,144,287,96,wcol)
3070 line(287,96,287,287,wcol)
3080 endfunc
3090 func drawsl1(wcol)
3100 /*左 1*/
3110 box(0,48,47,335,wcol)
3120 endfunc
3130 func drawsl2(wcol)
3140 /*左 2*/
3150 box(48,96,95,287,wcol)
3160 endfunc
3170 func drawsl3(wcol)
3180 /*左 3*/
3190 box(96,144,143,240,wcol)
3200 endfunc
3210 func drawsl1(wcol)
3220 /*右 1*/
3230 box(383,335,336,48,wcol)
3240 endfunc
3250 func drawsl2(wcol)
3260 /*右 2*/
3270 box(335,287,288,96,wcol)
3280 endfunc
3290 func drawsl3(wcol)
3300 /*右 3*/
3310 box(287,240,241,144,wcol)
3320 endfunc
```

図1

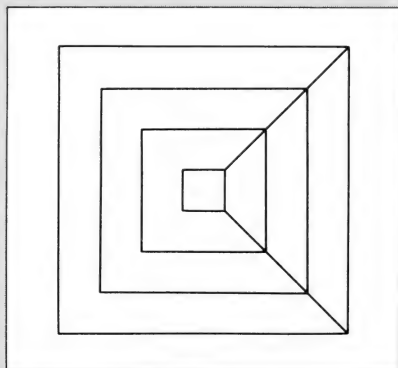
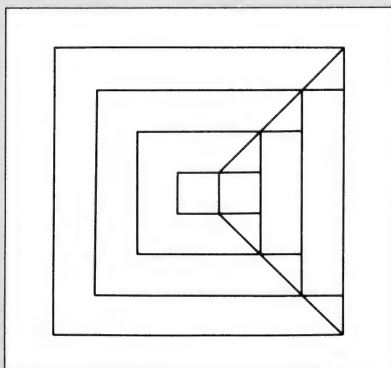


図2



X68000用 ©KONAMI

グラディウスⅢより **Sand Storm** Kashiwagi Katutoshi 柏木 勝利

X1/turbo用 ©システムサコム/Yonao Keishi

メタルサイトより **Into The Shadow** Takahashi Tetushi 高橋 哲史

やはりきたか、グラⅢ

X68000用にはグラディウスⅢからステージ1の“Sand Storm”をお送りしましょう。グラディウスといえば、X68000とは切っても切れない関係にあるシューティングゲームでしょう。その後継であるグラディウスⅢは、マニアのためにあったとまでいわれたゲームで、あまり普通の人がきではなかったようです。

さて、この作品は見てもらえばすぐにわかることとは思いますが、Yコマンドのてんこ盛りになっています。作品の完成度を高めるためにはしょうがないことかもしれませんが、やはり入力する人のことを考えるとあまりお勧めできません。西川善司さんのように、サブルーチンを作って展開するほうがスマートになると思いますよ。関数もきっちりと煮詰めれば、Yコマンドの行列と同等の効果は得られることでしょう。サンプリングだけでも変数に定義していたのが、せめてもの救いというものです。

まあ、それだけ凝っただけあって見事に採用となりました。プログラムを作る人も大変だったのでしょう。苦労話が同封の手紙につらつらと書き連ねてありました。プログラムは、よく見てみると使い回しがきく行があるようなので、そこいらへんを最初にチェックしてから入力すると、少しは楽になるかもしれません。

このリストを入力した人へのリクエストなのですが、ぜひOPMAではなく、OPMDで聴いてください。なぜなら、サンプリングされているドラム音が、すべてKORGの



グラディウスⅢ

M1からサンプリングされているOPMAに対し、OPMDではM1のほかに一部で専用のドラムマシンの音を用いているからなのです。そのせいか、OPMDで聴いたほうが曲により迫力が加わるようです。

ミッション・メタルサイト

先月号で、スタッフが作ればどんな機種でも毎月掲載できるんだぞ！ と書いたのは、皆さんを叱咤激励するつもりだったのです。ところが、その原稿を締め切りギリギリに持ち込んだ私を待っていたのは、お絵描きスタッフの高橋君から届いていた投稿だったのです……。

断っておきますが、頼んで作ってもらったではありません。あくまでも投稿扱いをしています。いままで投稿をしたことがある人ならわかると思いますが、きっと彼のもとにもOh!X特製の記念品が届くことでしょう。十分な審査のうえ、今月送られてきた曲のなかではいちばんよくできていたので採用になりました。

曲は、X68000専用シューティングゲーム“メタルサイト”のステージ1の曲、Into

木の葉も黄色くなって、もう街はすっかり秋の色。吹いてくる風もちょっぴり冷たくなってきました、みなさんお元気ですか？

さて、今月のLIVE inは、そんな落ち着いた情景とはうらはらに、ノリのいいゲームミュージック2本立てで攻めてみました。どちらも力作、ぜひ打ち込んで聴いてみてくださいね。



メタルサイト

The Shadowです。このゲームは全体的にレベルが高くて、グラフィックや操作性はもちろん、BGMもかなり秀逸なデキなのです。その移植とあれば十分に難しいと考えられます。まあ、音源の数を単純に比較してしまえば、X1のほうが有利ではありますが、やっぱりサンプリングの有無の影響は、結構大きいものなのでしょう。

ところがこの作品は、初めてMMLをさわった人が作ったとは思えないくらいによくできています。もともと、高橋君はお絵描き専門だったはず。ましてや「いままで音楽プログラムを作ったことはなかったんですよ」とは本人の弁。それでもここまでできるんですね。きっとよいアドバイザーがいて、意見してくれたんだと思います。皆さんも、作品を作ったらすぐに投稿しないで、友達や兄弟など、無理やりにも聴かせていろいろと評価してもらいましょう。意外と自分では気がつかなかった音のはずれなどが見つかったりしますよ。

皆さんも見よう見真似でもいいんです、チャレンジしてみてください。ひょっとすると来月号のLIVE inを飾るのは、あなたの作品かもしれませんよ。(S.K.)

リスト1 グラディウスⅢ

```
10 /*      save"GRADIUSIII.ST1
20 /*
30 /*      G R A D I U S I I I
40 /*
50 /*      S a n d   S t o r m ( Stage1 . 砂 塵 )
60 /*
70 /*      作曲・編曲 コナミ矩形波倶楽部
80 /*
90 /*      P R O G R A M E D   B Y   柏 木 勝 利
```

```
100 /*
110 /*
120 m_init()
130 /*
140 str pd(30)[256]:pd(30)="{*}"
150 char po(255),v(4,10),vo(4,9)
160 /*
170 str p0="y3,0", p1="y3,1", p2="y3,2", p3="y3,3"
180 str bd="y2,23",sd="y2,17",hc="y2,65",ho="y2,66"
```



```

190 str t1="y2,28",t2="y2,29",t3="y2,30",t4="y2,31"
200 str c1="y2,3", cc="y2,5"
210 /*
220 for i=1 to 8
230   m_alloc(i,3000)
240   m_assign(i,i)
250 next
260 /*
270 for i=1 to 8
280   m_trk(i,"[d.c.][coda]")
290 next
300 /*
310 VOI()
320 MML1()
330 MML2()
340 m_play()
350 end
360 /*
370 /* TRACK SET
380 /*
390 func trk(t)
400   c=0
410   while po(c)<>255
420     m_trk(t,po(c))
430     c=c+1
440   endwhile
450 endfunc
460 /*
470 /* VOICE SET
480 /*
490 func vset(no)
500   v(0,0)=(vo(4,1)*8)+vo(4,0)
510   v(0,1)=15
520   v(0,9)=3
530   for x=0 to 3
540     for y=0 to 9
550       v(x+1,y)=vo(x,y)
560     next
570   next
580   m_vset(no,v)
590 endfunc
600 /*
610 /* GRADIUS III VOICE DATA
620 /*
630 func VOI()
640 /*
650 /*
660 /* AR DR SR RR SL OL KS ML DT1 DT2
670 vo={ 31, 11, 0, 10, 11, 34, 0, 0, 7, 0,
680        31, 10, 1, 10, 12, 45, 0, 8, 5, 0,
690        31, 10, 2, 9, 13, 32, 0, 0, 2, 0, /* C
ON FBL
700 31, 2, 0, 12, 8, 0, 0, 1, 3, 0,
2, 7]
710 vset(70)
720 /*
730 /*
740 /* AR DR SR RR SL OL KS ML DT1 DT2
750 vo={ 18, 8, 0, 10, 2, 29, 0, 4, 4, 0,
760        28, 2, 1, 12, 15, 3, 0, 4, 4, 0,
770        28, 2, 1, 12, 15, 3, 0, 4, 4, 0, /* C
ON FBL
780 28, 2, 1, 12, 15, 4, 0, 4, 4, 0,
5, 7]
790 vset(71)
800 /*
810 /*
820 /* AR DR SR RR SL OL KS ML DT1 DT2
830 vo={ 20, 4, 0, 8, 4, 29, 0, 4, 3, 0,
840        20, 4, 0, 10, 12, 1, 0, 8, 3, 0,
850        20, 4, 0, 8, 4, 14, 0, 4, 7, 0, /* C
ON FBL
860 20, 4, 0, 10, 12, 2, 0, 4, 7, 0,
4, 5]
870 vset(72)
880 /*
890 /*
900 /* AR DR SR RR SL OL KS ML DT1 DT2
910 vo={ 20, 8, 0, 10, 2, 30, 0, 4, 0, 0,
920        30, 2, 0, 12, 15, 4, 0, 4, 0, 0,
930        30, 2, 0, 12, 15, 4, 0, 4, 0, 0, /* C
ON FBL
940 30, 2, 0, 12, 15, 4, 0, 4, 0, 0,
5, 7]
950 vset(73)
960 /*
970 /*
980 /* AR DR SR RR SL OL KS ML DT1 DT2
990 vo={ 31, 24, 0, 12, 15, 35, 0, 14, 7, 0,
1000 31, 13, 0, 8, 15, 0, 0, 2, 7, 0,
1010 31, 20, 0, 4, 15, 36, 0, 13, 3, 0, /* C
ON FBL
1020 31, 12, 0, 5, 15, 0, 0, 1, 3, 0,
4, 3]
1030 vset(74)
1040 /*
1050 /*
1060 /* AR DR SR RR SL OL KS ML DT1 DT2
1070 vo={ 22, 0, 0, 0, 24, 0, 4, 7, 0,
1080 20, 1, 0, 6, 2, 0, 0, 4, 7, 0,
1090 22, 0, 0, 0, 45, 0, 12, 3, 0, /* C
ON FBL
1100 20, 1, 0, 6, 2, 0, 0, 4, 3, 0,
4, 7]
1110 vset(75)
1120 /*
1130 /*
1140 /* AR DR SR RR SL OL KS ML DT1 DT2

```

```

1150 vo={ 31, 28, 0, 15, 15, 19, 0, 1, 3, 2,
1160 31, 18, 11, 15, 15, 0, 0, 1, 7, 0,
1170 31, 18, 11, 12, 15, 0, 0, 1, 7, 0, /* C
ON FBL
1180 31, 18, 11, 15, 15, 0, 0, 1, 3, 0,
5, 6]
1190 vset(76)
1200 /*
1210 /*
1220 /* AR DR SR RR SL OL KS ML DT1 DT2
1230 vo={ 31, 8, 0, 4, 2, 2, 0, 10, 2, 1,
1240 31, 18, 3, 15, 13, 42, 0, 0, 0, 2,
1250 31, 20, 12, 12, 14, 22, 0, 0, 3, 3, /* C
ON FBL
1260 31, 6, 3, 15, 15, 0, 0, 1, 0, 0,
3, 7]
1270 vset(77)
1280 /*
1290 /*
1300 /* AR DR SR RR SL OL KS ML DT1 DT2
1310 vo={ 31, 28, 0, 1, 5, 20, 0, 3, 3, 0,
1320 31, 16, 0, 8, 5, 4, 0, 7, 7, 1,
1330 31, 28, 0, 1, 5, 14, 0, 7, 7, 2, /* C
ON FBL
1340 31, 16, 0, 8, 5, 0, 0, 11, 3, 3,
4, 7]
1350 vset(78)
1360 /*
1370 /*
1380 /* AR DR SR RR SL OL KS ML DT1 DT2
1390 vo={ 29, 1, 0, 2, 1, 26, 0, 1, 3, 0,
1400 31, 1, 0, 10, 2, 0, 0, 1, 3, 0,
1410 29, 1, 0, 2, 1, 32, 0, 1, 7, 0, /* C
ON FBL
1420 31, 1, 0, 10, 2, 0, 0, 1, 7, 0,
4, 7]
1430 vset(79)
1440 /*
1450 /*
1460 /* AR DR SR RR SL OL KS ML DT1 DT2
1470 vo={ 20, 0, 0, 0, 0, 24, 0, 4, 7, 0,
1480 15, 1, 0, 6, 2, 3, 0, 4, 7, 0,
1490 20, 0, 0, 0, 0, 45, 0, 12, 3, 0, /* C
ON FBL
1500 15, 1, 0, 6, 2, 3, 0, 4, 3, 0,
4, 7]
1510 vset(80)
1520 /*
1530 /*
1540 /* AR DR SR RR SL OL KS ML DT1 DT2
1550 vo={ 28, 2, 0, 2, 1, 31, 0, 2, 3, 0,
1560 26, 1, 1, 6, 2, 6, 0, 6, 3, 0,
1570 28, 2, 0, 2, 1, 24, 0, 2, 7, 0, /* C
ON FBL
1580 26, 1, 1, 6, 2, 6, 0, 6, 7, 0,
4, 7]
1590 vset(81)
1600 /*
1610 endfunc
1620 /*
1630 /* MML DATA - 1
1640 /*
1650 func MML1()
1660 m_tempo(161)
1670 /*
1680 /* * * * * *
1690 pd(0)="0710v12202013q8p3y48,8"
1700 pd(1)="g4.&:y48,193g-&y48,123g-&y49,193g-&y48,8g&y48,78g&
y48,148g&y48,78g&y48,8g&:y48,183g-&y48,183g-&y48,183g-&y48,8g&y
48,88g&y48,168g&y48,88g&y48,8g072g12b12d12
1710 pd(2)="g4.&:y48,223g-&y48,183g-&y48,223g-&y48,8g&y48,48g
&y48,88g&y48,48g&y48,8g&:y48,213g-&y48,163g-&y48,213g-&y48,8g&y
48,58g&y48,108g&y48,58g&y48,8g071
1720 pd(3)="a4.&:y48,193a-&y48,123a-&y49,193a-&y48,8a&y48,78a&
y48,148a&y48,78a&y48,8a&:y48,183a-&y48,183a-&y48,183a-&y48,8a&y
48,88a&y48,168a&y48,88a&y48,8a072a12c12f12
1730 pd(4)="a4.&:y48,223a-&y48,183a-&y48,223a-&y48,8a&y48,48a
&y48,88a&y48,48a&y48,8a&:y48,213a-&y48,163a-&y48,213a-&y48,8a&y
48,58a&y48,108a&y48,58a&y48,8a071
1740 pd(5)="b4.&:y48,193b-&y48,123b-&y49,193b-&y48,8b&y48,78b&
y48,148b&y48,78b&y48,8b&:y48,183b-&y48,183b-&y48,183b-&y48,8b&y
48,88b&y48,168b&y48,88b&y48,8b072b12c12f12
1750 pd(6)="b4.&:y48,223b-&y48,183b-&y48,223b-&y48,8b&y48,48b
&y48,88b&y48,48b&y48,8b&:y48,213b-&y48,163b-&y48,213b-&y48,8b&y
48,58b&y48,108b&y48,58b&y48,8b071
1760 pd(7)="c4.&:y48,193c-&y48,123c&y49,193c&y48,8c&y48,78c&
y48,148c&y48,78c&y48,8c&:y48,183c-&y48,183c&y48,183c&y48,8c&y48
,88c&y48,168c&y48,88c&y48,8c072c12f12a12c
1770 pd(8)="c4.&:y48,223c-&y48,183c&y48,223c&y48,8c&y48,48c&
y48,88c&y48,48c&y48,8c&:y48,213c-&y48,163c&y48,213c&y48,8c&y48
,58c&y48,108c&y48,58c&y48,8c
1780 pd(9)="0710v11702g2d2&:y48,193d-&y48,123d-&y48,193d-&y4
8,8d&y48,78d&y48,148d&y48,78d&y48,8d&:y48,183d-&y48,183d-&y48,183d-&y48,8d&y48,88d&y48,168d&y48,88d&y48,8d
1790 pd(10)="d6c6b6c4&c6g12&g2&:y48,183g-&y48,103g-&y48,183
g-&y48,8g&y48,88g&y48,168g&y48,88g&y48,8g&:y48,173g-&y48,83g-&y
48,173g-&y48,8g&y48,98g&y48,188g&y48,98g&y48,8g
1800 pd(11)="c6b6c6&a2&:y48,183a-&y48,103a-&y48,183a-&y48,8
a&y48,88a&y48,168a&y48,88a&y48,8a&:y48,173a-&y48,83a-&y48,173a1
2a-cy48,8c6d6
1810 po={ 0,1,2,3,4,5,6,7,8,9,10,9,11,255 }
1820 trk(1)
1830 /*
1840 /* * * * * *
1850 pd(0)="0710v11002013q8p3y49,32r12"
1860 pd(1)="g4.&:y49,217g-&y49,147g-&y49,217g-&y49,32g&y49,102
g&y49,172g&y49,102g&y49,32g&:y49,207g-&y49,127g-&y49,207g-&y49,
32g&y49,112g&y49,192g&y49,112g&y49,32g072g12b12c12

```


[illegible]

80 · Oh! X 1990.12


```

y49,172g&y49,102g&y49,32g":pd(10)=pd(9)+"f+6d6>a6":pd(11)=pd(9)+
" g6cebg6
3220 pd(12)=""f+6g6a4.&y49,217a-&y49,147a-&y49,217a-&y49,32a&y49
,102a&y49,172a&y49,102a&y49,32a&y49,207a-&y49,127a-&y49,207a-&y4
9,32a&y49,102a&y49,1720lla
3230 trk(2)
3240 /*
3250 endfunc
3260 /*
3270 /* こゝから先を削除。
3280 /*

```

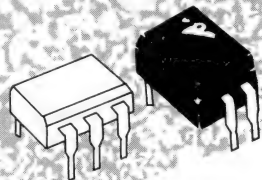
リスト2 メタルサイト

```

890 D=STRINGS(2,"01140C03139E")+ "R4E16E16E16E16"
890 E=STRINGS(2,"01140C13903E")+ "E16E16E16E16E16E8"
900 PLAY " :R1R1R1R1R1R1";
910 PLAY A;
920 PLAY "[ "+STRINGS(4,B);
930 PLAY STRINGS(3,B);
940 PLAY C;
950 FOR Z=0 TO 1
960 PLAY B+B+B+B;
970 PLAY B+B+B+D;
980 PLAY B+B+B+B;
990 PLAY B+B+B+B;
1000 PLAY B+B+B+B+B;
1010 PLAY D;
1020 NEXT
1030 PLAY D+D+D+E+"I";
1040 'I I I I I MELODY
1050 A="O211V11"+STRINGS(16,"F+1&")
1060 B="13R2R8L8C+EF+G+F+16E16E2,"
1070 C="R4R8C+EF+G+F+G+8.+A8.B&B2"
1080 D="R2R8C+EF+G+A16G+16&G+2EC+"
1090 E="<B8.>C+8.+F+8&F+2&F+1"
1100 F="F+2.G+AB8.>C+8.<F+8&F+4.E"
1110 G="F+2.G+AB8.>C+8.F+8&F+2<"
1120 H="O211V11"+STRINGS(12,"F+1&")
1130 PLAY ":O2P1"+A+"[";
1140 FOR Z=0 TO 1
1150 PLAY "O6V16"+B+C+D+E;
1160 PLAY "O6"+B+C+D+F+F+F+G;
1170 NEXT
1180 PLAY H+"]";
1190 'I I I I I MELODY SUB
1200 PLAY " :O211K20P2R03"+A+"[";
1210 FOR Z=0 TO 1
1220 PLAY "K1006V16"+B+C+D+E;
1230 PLAY "O6"+B+C+D+F+F+F+G;
1240 NEXT
1250 PLAY H+"]";
1260 PLAY " :O2K5Y7,28"+A+"[";
1270 FOR Z=0 TO 1
1280 PLAY "K3O3V11"+B+C+D+E;
1290 PLAY B+C+D+F+F+F+G;
1300 NEXT
1310 PLAY H+"]";
1320 PLAY " :O2K10R03"+A+"[";
1330 FOR Z=0 TO 1
1340 PLAY "K003V11"+B+C+D+E;
1350 PLAY B+C+D+F+F+F+G;
1360 NEXT
1370 PLAY H+"]";
1380 'I I I I I HI-HAT
1390 A="V1202L16S0,0,0,0=1'y6,30Q5"
1400 B="C8C8CCCCCcy6,12cy6,30CC8C8"
1410 C="C8C8CCCCCcy6,12cy6,30CR4"
1420 D="O3Q8y6,5C8Q5y6,30O2C8CCCCCcy6,12Cy6,30CC8C8"
1430 PLAY " :"+STRINGS(8,"R1");
1440 PLAY "[ "+A+B+B+B+C;
1450 PLAY D+B+B+C;
1460 FOR Z=0 TO 1
1470 PLAY STRINGS(7,B)+C;
1480 PLAY D+STRINGS(6,B);
1490 PLAY STRINGS(6,B)+C;
1500 NEXT
1510 PLAY B+B+B+"C8C8CCCCR2"+"]"
1520 END
1530 LABEL "INST"
1540 MEMS(&HB190,36)=HEXCHR$("F5 00 31 22 40 62 14 1B 1B 11 14 1
4 14 14 00 05 00 00 00 01 00 00 51 A8 F8 F5 00 00 00 00 00 00 80
00 02 00") '11 MELODY 1
1550 MEMS(&HB184,36)=HEXCHR$("F3 00 45 42 41 42 16 1B 0C 9C 1
C 1C 18 02 02 00 08 04 06 04 04 04 F6 00 00 00 00 00 CC 80
00 02 00") '12 CHORD 1
1560 MEMS(&HB1D8,36)=HEXCHR$("DA 50 20 32 31 70 27 1A 23 00 DE D
F 5E 4D 01 01 01 82 00 00 00 11 15 15 04 00 88 80 80 0C CD 8A
00 02 00") '13 MELODY 2
1570 MEMS(&HB1FC,36)=HEXCHR$("F8 00 02 01 00 00 25 1E 1B 07 94 5
4 54 01 06 06 06 07 06 06 08 A4 A4 14 F6 00 00 00 00 00 00 80
00 02 00") '14 CHORD 2
1580 MEMS(&HB244,36)=HEXCHR$("20 00 66 65 60 61 1C 3A 16 00 DF D
F 9F 9F 06 05 08 05 06 05 05 07 24 14 14 F4 00 00 00 00 00 00 80
00 00 00") '16 BASS 1
1590 MEMS(&HB268,36)=HEXCHR$("F8 00 36 35 30 31 1D 2F 1D 00 DF D
F 9A 9F 07 06 09 88 07 06 06 04 29 19 39 80 00 00 00 00 C8 80
00 02 00") '17 BASS 2
1600 MEMS(&HB6E8,36)=HEXCHR$("FC 00 0C 01 00 00 00 07 11 07 1F S
C 5E 5C 00 0D 11 0D C0 00 80 00 01 F7 D9 F7 00 00 00 00 F4 C8 80
00 02 00") 'I39 SnareDrum2
1610 MEMS(&HB70C,36)=HEXCHR$("80 00 01 0E 00 00 00 00 00 00 1E 1
E 1E 1D 1A 1C 0F 8C 40 C0 40 00 FD FE F8 F8 00 00 00 00 D0 C8 80
00 02 00") 'I40 Bass Drum1
1620 RETURN

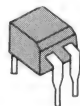
```


A/Dコンバータその3

Misawa Kazuhiko
三沢 和彦

A/Dコンバータの製作もいよいよ3回目。今回はその応用編ということで、先月作ったボリューム1個の簡易アナログジョイスティックを使ってブロック崩しを作ってみましょう。プログラムはX-BASICで書いたものをコンパイルして使用します。

先月までに、簡単な回路で応用性抜群のA/Dコンバータボードを製作し、まず最初の応用として簡易アナログジョイスティックを紹介してみました。今回は、アナログジョイスティックを使った簡単なゲームプログラムを組んでみましょう。内容は、昔年つかしの「ブロック崩し」です。



アナログジョイスティックの使い方

まずは、A/Dコンバータとボリューム1個の組み合わせでアナログジョイスティックとして活用する方法を考えてみましょう。図1が簡易アナログジョイスティックの全回路図です。

ボリュームには1本の抵抗の両端にある端子2個とその抵抗の途中に接触している中間端子1個とがあります。ボリュームの

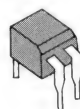
固定抵抗の両端に本体電源からの+5VとGNDがつながっていて、ボリュームの中間端子がA/Dコンバータの電圧入力端子につながっています。

この中間端子からは、ボリュームの両端の電圧以下の電圧を連続的に取り出すことができます。それは、この中間端子がボリュームの抵抗線にじかに接触していて、ボリュームを回すことによって、両端の固定端子からの距離を連続的に変化させることができるからです。そして、両端のそれぞれの端子までの長さにしたがって内分された電圧が取り出されることになります(図1参照)。

したがって、ボリュームのツマミの位置によって中間端子の電圧が変わることになります。先月掲載のサンプルプログラムでは画面上に現れる数字がその中間端子の電

ーチンになっています。これは単に読み出してきたデータをX座標としてパドルを定義したスプライトを移動させているだけです。もちろんA/Dコンバータからの読み出しは先月のルーチンとまったく同じものです。

このように、アナログジョイスティックとはいっても実は256段階のデジタル変化ですが、コンパイル後のプログラムでは、画面上のパドルの動きはずいぶんスムーズであることが実感できると思います。



ブロック崩しプログラミング

ゲームのプログラミング自体は簡単で、むしろ皆さんのほうがデザインセンスが優れていると思います。今回のサンプルゲームは、自作の簡易アナログジョイスティックでパドルを左右に動かし、ボールを返しながら、ブロックを消していきます。1ステージをクリアするとブロックの数が増え、パドルが前に進んでいきます。プレイヤーは持ち数が3で、これも1ステージクリアごとにひとつずつ増えていきます。

ボールとブロックとの衝突判断も若干不自然なところが残っていますが、ステージをクリアし続けるとパドルがブロックよりも前に出てしまうバグがありますが、X-BASICでの改良は皆さんにお任せします。ただし、コンパイルしないとまったくゲームになりません。コンパイルは、

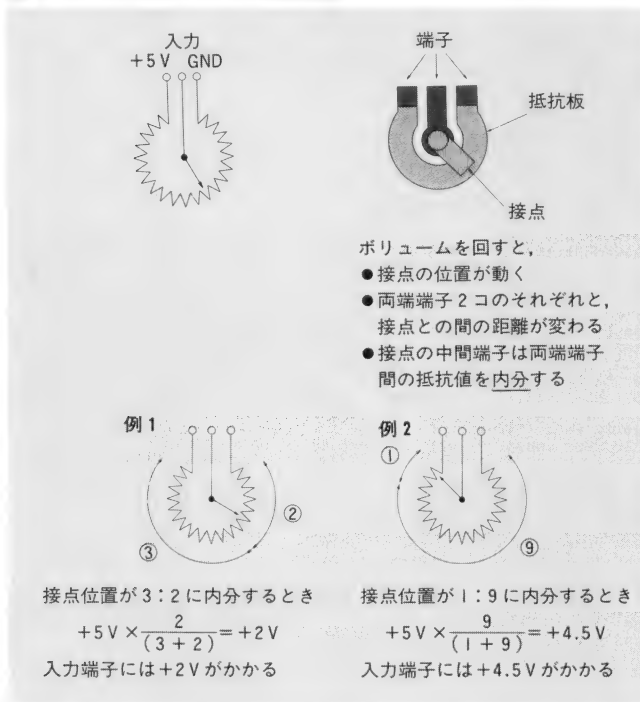
CC GAME.BAS IODRV.O

とすれば、実行ファイルGAME.Xが生成されます。なお、XCのver.2.0を使った場合には、コンパイルの途中で、

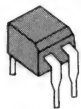
ERROR 15: function return value mismatch

ERROR 16: argument type mism

図1 アナログジョイスティックの回路図



atch
の2種類の警告(Warning)が出ますが、
これは無視してかまいません。



A/Dコンバータ用外部関数

このA/Dコンバータは次のセンサ応用編でも使います。それだけ応用範囲の広い外部機器なのです。X68000からコントロールするときはread関数、start関数、clock関数をまったく同じ形で使えるのですが、これだけ汎用性があるのですから、その都度X-BASICのプログラムのなかで定義しないで、このread関数もマシン語で外部関数にしてしまうと便利です。

というわけで、おなじみのioinp関数、ioout関数に加えて、adread関数を新しく付け加えました。リスト2は以前載せた外

部関数のソースリストにそのまま加えたもので、ioinp関数、ioout関数の部分はまったく変更ありませんので、以前と同様に使えます。付け加えたadread関数の部分は、

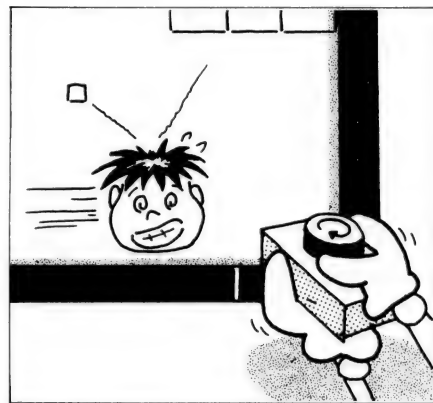
(追加した部分)

のように囲ってあります。このソースファイルを新たに、

CC IODRV.S

でコンパイルし、出来上がったIODRV.XをIODRV.FNCにリネームしてやれば、BASICでのread関数とまったく同じ使い方ができるようになります(readをadreadに書き直すだけ!)

この外部関数を使えば、次のプログラミングはたいへん短いものになりますが、どうもアセンブラは苦手で最初のiodrv.fn



cだけでも大変だったという初心者の方は、今までどおりBASICのread関数を使ってもかまいません。

以上、自作A/Dコンバータ基板の簡単な応用を紹介してみました。次回からは、より身近で、より実用的な回路に挑戦してみたいと思いますので、楽しみに。

リスト1 ブロック崩し

```
10 /* save "d:\basic\game.bas
20 /* save@d:\basic\game.doc
30 /******
40 /*
50 /* A/Dコンバータ&簡易ジョイスティック
60 /* サンプルプログラム
70 /*
80 /* なつかしのブロック崩し
90 /*
100 /* 1990.10.10. K.Misaka
110 /*
120 /******
130 screen 1,1,1,1
140 vpage(7)
150 console,,0 : color 6
160 int v=0,cs=0,di=0,sgl,sign
170 int ch=0
180 int bx,by,sx=0,sy,px,xx,spd=2
190 int rwall=250,lwall=10,uwall=10,bdiam=8
200 int bhight=8,bwidth=16,brow=1,bcol=15
210 int ofs,ppos=466,bases=32
220 ofs=bwidth/2 : bhight=4*spd
230 int col,rak,colnxt,ravnxt,pl
240 int stage=0,count=0,total=0,maxplayers=3,player
250 player=maxplayer-1
260 int flg,endflg=1,hscore=0
270 /*
280 dim int block(17,50)
290 dim int cl(5)= {3,5,7,9,11,13}
300 dim float gpat(375)
310 locate 42,6: print "HIGH SCORE"
320 locate 42,8: print "0"
330 locate 42,12: print "SCORE"
340 locate 42,18: print "STAGE"
350 locate 42,24: print "PLAYER"
360 /*
370 sp_init()
380 sp_off(0,127)
390 spdef()
400 sp_disp(1)
410 wall()
420 back_ground()
430 /*
440 /*メインルーチン
450 /*
460 while endflg
470 initialize()
480 stage_clear()
490 one_play()
500 game_over()
510 endwhile
520 end
530 /*
540 /*パラメータ初期化
550 /*
560 func initialize()
570 stage=0:count=0:total=0
580 player=maxplayer-1
590 ppos=466
600 locate 42,14: print "0"
610 endfunc
620 /*
630 /*ゲームオーバー・リプレー処理
640 /*
650 func game_over()
660 locate 11,12: print "GAME OVER!"
670 locate 4,18: print "HIT SPACE KEY FOR REPLAY!"
680 if inkey$="" then endflg=1 else endflg=0
690 locate 11,12: print " "
700 locate 4,18: print " "
710 endfunc
```

```
720 /*
730 /*ワンプレー
740 /*
750 func one_play()
760 for pl=1 to player
770 message()
780 while move_ball()=1
790 if by>=base+uwall-bdiam/2*(sgn(sy)+1) and by<=base+uwa
l+bhight*brow+(sy>0) and ((by-base-uwall) mod bhight)=0 then {
800 check_block() }
810 move_paddle()
820 endwhile
830 sp_off(1)
840 next
850 endfunc
860 /*
870 /*プレーヤー
880 /*
890 func message()
900 locate 13,12: print "READY?"
910 for i=1 to 40000: next
920 locate 13,12: print " "
930 bx=(rwall+lwall-bdiam)/2 : by=uwall+base+bhight*(brow+2)
940 sx=0 : sy=spd*4
950 sp_move(1,bx,by,1)
960 locate 42,26: print using "*****":player=pl+1
970 endfunc
980 /*
990 /*パドル操作
1000 /*
1010 func move_paddle()
1020 x=read(ch)
1030 if x<lwall+ofs then x=lwall+ofs
1040 if x>rwall-ofs then x=rwall-ofs
1050 if abs(xx-x)>1 then xx=x : px=xx-ofs
1060 sp_move(0,px,ppos+bdiam,0)
1070 endfunc
1080 /*
1090 /*ボール運動の判定
1100 /* (戻り値) 1:インプレー
1110 /* 0:ミス
1120 /*
1130 func int move_ball()
1140 bx=bx+sx : by=by+sy
1150 if by>511 then return(0)
1160 if bx<lwall then bx=lwall : sx=-sx
1170 if bx>rwall-bdiam then bx=rwall-bdiam : sx=-sx
1180 if by<lwall then by=lwall : sy=-sy
1190 sp_move(1,bx,by,1)
1200 if by>=ppos and by<=ppos+bhight then check()
1210 return(1)
1220 endfunc
1230 /*
1240 /*パドルにヒットしたかの判定
1250 /*
1260 func check()
1270 if bx>px-bdiam and bx<px+bwidth then {
1280 by=ppos : sy=-sy : sx=int((bx-px-2+sgn(bx-px-4))/4)*sp
d : beep }
1290 endfunc
1300 /*
1310 /*ブロックにヒットしたかの判定
1320 /*
1330 func check_block()
1340 col=(bx+bdiam/2-lwall)*bwidth+1 : raw=(by-uwall-base)*bh
ight+1+sgn(sy)
1350 colnxt=col+sgn(sx) : ravnxt=raw+sgn(sy)
1360 if block(col,raw)=1 then { erase_block(col,raw,1) } else
{
1370 if (bx+bdiam/2+sx-lwall)*bwidth+1<col and block(colnxt
,raw)=1 then { erase_block(colnxt,raw,0)
```



```

69: *
70: ptr_token:      dc.b  'ioinp',0
71:                dc.b  'ioout',0
72: *****
73:                dc.b  'adread',0
74: *****
75:
76:                dc.b  0
77:
78:                .even
79:
80: *
81: *パラメータテーブル
82: *
83: ptr_param:      dc.l  ioinp_par
84:                dc.l  ioout_par
85: *****
86:                dc.l  adread_par
87: *****
88:
89: *
90: *パラメータ I D テーブル
91: *
92: ioinp_par:      dc.w  int_ret
93: ioout_par:      dc.w  int_val
94:                dc.w  void_ret
95: *****
96: adread_par:     dc.w  int_val
97:                dc.w  int_ret
98: *****
99:
100: *
101: *実行アドレステーブル
102: *
103: ptr_exec:       dc.l  ioinp_exec
104:                dc.l  ioout_exec
105: *****
106:                dc.l  adread_exec
107: *****
108:
109:                .even
110:
111: *****
112: *
113: *                定義関数ルーチン
114: *
115: *****

116: *
117: *
118: *データ入力関数 ioinp()
119: *
120: porta          equ    $e9a001
121:
122: ***実行アドレス
123: ioinp_exec:     bsr    _ioinp
124:                move.l #0,d0
125:                rts
126:
127: ***メインルーチン
128: _ioinp:
129:
130: ***スーパーバイザモードに入る
131:                clr.l  -(sp)
132:                dc.w   _SUPER
133:                addq.l #4,sp
134:                move.l d0,spbuf
135:
136: rd_ok:          clr.l  d1
137:
138: ***ジョイスティックポートから読みだし
139:                move.l #porta,d2
140:                movea.l d2,a3
141:                move.b (a3),d1
142:
143: ***ユーザーモードに戻る
144:                move.l spbuf,-(sp)
145:                dc.w   _SUPER
146:                addq.l #4,sp
147:
148: ***戻り値をバッファに格納
149: rd_ready:       move.l d1,int_data
150:                lea.l  retdat,a0
151:                move.l d1,d0
152:
153:                rts
154:
155: *
156: *データ出力関数 ioout(data)
157: *
158: portc           equ    $e9a005
159:
160: ***実行アドレス
161: ioout_exec:
162:
163: ***引数をスタックに積み替え
164:                move.l 12(sp),d1
165:                move.l d1,-(sp)
166:
167:                bsr    _ioout
168:                addq.l #4,sp
169:
170:                move.l #0,d0
171:                rts
172:
173: ***メインルーチン
174: _ioout:
175:
176: ***スーパーバイザモードに入る
177:                clr.l  -(sp)
178:                dc.w   _SUPER
179:                addq.l #4,sp
180:                move.l d0,spbuf
181:
182: ***ジョイスティックポートへ書き込み
183: wr_ok:          move.l #portc,d2
184:                movea.l d2,a3
185:                move.l 4(sp),d1

```

```

186:                move.b d1,(a3)
187:
188: ***ユーザーモードに戻る
189:                move.l spbuf,-(sp)
190:                dc.w   _SUPER
191:                addq.l #4,sp
192:
193:                rts
194:
195: *****
196: *
197: *A / D コンバータ読み込み関数 adread(ch)
198: *
199: high           equ    %0
200: low            equ    %10000000
201:
202: ***実行アドレス
203: adread_exec:
204:
205: ***引数をスタックに積み替え
206:                move.l 12(sp),d1
207:                move.l d1,-(sp)
208:
209:                bsr    _adread
210:                addq.l #4,sp
211:
212:                move.l #0,d0
213:                rts
214:
215: ***メインルーチン
216: _adread:        move.l #%1000000,-(sp)
217:                bsr    _ioout
218:                addq.l #4,sp
219:
220: ***A / D 変換スタート
221:                move.l #high,-(sp)
222:                bsr    clock
223:                addq.l #4,sp
224:
225:                move.l #high,-(sp)
226:                bsr    clock
227:                addq.l #4,sp
228:
229:                move.l 4(sp),d0
230:                dbra   d0,ch1
231:
232: ***チャンネル0 指定
233: ch0:           move.l #low,-(sp)
234:                bsr    clock
235:                addq.l #4,sp
236:                bra    read
237:
238: ***チャンネル1 指定
239: ch1:           move.l #high,-(sp)
240:                bsr    clock
241:                addq.l #4,sp
242:
243: ***シリアルデータ読み込み
244: read:          move.l #7,d5
245:                clr.l  d6
246:
247: loop:          move.l #low,-(sp)
248:                bsr    clock
249:                addq.l #4,sp
250:
251:                bsr    _ioinp
252:                lsl.l  #1,d6
253:                andi.l #1,d0
254:                add.l  d0,d6
255:
256:                dbra   d5,loop
257:
258: ***戻り値をバッファに格納
259: ready:         move.l d6,int_data
260:                lea.l  retdat,a0
261:
262:                move.l #%0,-(sp)
263:                bsr    _ioout
264:                addq.l #4,sp
265:
266:                move.l d6,d0
267:
268:                rts
269:
270: ***クロックルーチン(スタックにデータセット)
271: clock:         move.l 4(sp),d3
272:                addi.l #%1000000,d3
273:                move.l d3,d4
274:                addi.l #%10000,d4
275:
276:                move.l d3,-(sp)
277:                bsr    _ioout
278:                addq.l #4,sp
279:
280:                move.l d4,-(sp)
281:                bsr    _ioout
282:                addq.l #4,sp
283:
284:                move.l d3,-(sp)
285:                bsr    _ioout
286:                addq.l #4,sp
287:
288:                rts
289: *****
290:
291: *
292: *スタックバッファ
293: *
294: spbuf          ds.l  1
295:
296: *
297: *戻り値格納バッファ
298: *
299: retdat:        dc.w   0
300:                dc.l   0
301: int_data:       dc.l   0
302:                end
303:
304:

```


アナログジョイスティックの製作

Ishigami Tatsuya 石上 達也

「サイバースティック」を作る。以前ラジコンのプロポからの出力をアナログジョイスティックとして利用する記事がありましたが、今度はスティック部まで自作してみようというもの。はてさて、どうなりますやら。

以前、アナログスティックの製作として、1990年5月号に栗野氏によるものが、発表されました。ラジコンのプロポを入力装置に用い、受信機とインタフェイス回路を接続するという、たいへん独特なものでした。今回、この方法とはまったく別にZ80CPUを用いたコンピュータ制御のアナログスティックを製作したいと思います。

ここでは手持ちのもの、すなわち、PC-8801mkII+S-OS、ROMライター、その他工具一式、Z80の知識……を最大限に生かして「できるだけ安く」を目標にしていきます。よってこれらの資材、特にROMライターがない人はかえって手間がかかりますので、栗野氏のCPUを使わない制御回路に積み換えるなど工夫するといいでしょう。

さて、X68000のサイバースティックは「インテリジェントコントローラ」です。この「インテリジェント」というのがクセモノで、4ビットマイコンを使って多彩な機能をまとめているわけです。

我々に馴染みのあるCPU、Z80は8ビット。これも組み込み制御用に多用されているCPUです。おまけに非常に入手しやすく、プログラム開発も容易です。性能にも問題ありません。4ビットでできて8ビットでできないわけではないでしょう。

世のマイコン制御やインテリジェント家電機器の多くは4/8ビットマイコンで制御されています。これらをZ80で代用していくというのは結構面白い試みじゃないでしょうか。

回路について

図1がその回路です。よく見るとRAMがないのがわかると思います。これは制御プログラムを（やっとの思いで）ワークエリアを使わなくても動くようにしたためです（すなわち、LD(nnnn), AはおろかPUSH&POP, CALL&RETも使っていない）。

HM6116P-3が200円で買える時代に、

なんのために無駄なことをやっているんだ！ と思うかもしれませんが、26ピンのLSIを取り付けるのには、最低52カ所のハンダ付けをしなければなりません。また、RAMを省略することによって、メモリ関係のアドレスデコーダを省略することができます。

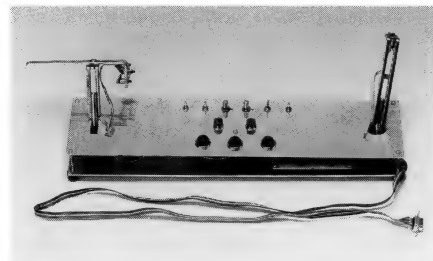
クロック発振回路は、もっと安価にできないこともないのですが、高価な測定器を持っていないので確実に発振が保証されているクロックオシレータを使用しました。まあ一応、縁起ものだと思ってください。

そして、もうひとつの縁起ものが、リセット素子のPST518です。リセット素子というのは、電源がふらついたときに、CPUなどに対して、リセット信号を出すものです。この部分も、抵抗とダイオードだけで済ませることも可能なのですが、1個50円でPST518が売っていましたので取り付けておきました。

CPUには、皆さんお馴染みのザイログZ80を用いています。もうこれは、説明する必要がないでしょう。X1やMZシリーズに搭載されているアレです。今回はクロック4MHzで動作させています。これにRAM48KバイトをつけたらMZ-80Kの約2倍のパフォーマンスを持つことになります。うーん技術の進歩に感謝感謝！

ROMの2764は、アクセスタイムが200ms以下のものを使ってください。ここらへんの回路はえらくシンプルにできました。やはり、制御ソフトのほうでまったくワークエリア（やスタック領域）を使用していないためです。そのため、アドレスデコーダなどが不要となり、ほとんどCPUと直結しています。

なぜ、ROMとして2764を使ったのかというと、理由は簡単で私が2764/128のROMライターしか持っていなかったからです。2732とか2716も書き込めるROMライターを持っている方は、そちらを使うとさらに安くあがるでしょう。回路はほとんど変更あ



どことなくサイバーな仕上がり

りません。回路図のように、ただCPUとROMを直結するだけです。

LS138は、I/O空間用のアドレスデコーダです（さすがにI/O空間用のデコーダは省略できない）。8255AとADC0809の割り振りを行っています。ちなみに、8255Aに対しては、A2のデコードを行っていないので一部影が出てきます。具体的には以下のように割り振っています。

- 80_H 8255のポートA
- 81_H 8255のポートB
- 82_H 8255のポートC
- 83_H 8255のコントロールワード
- 84_H 80_Hの影
- 85_H 81_Hの影
- 86_H 82_Hの影
- 87_H 83_Hの影
- 88_H ADC0809のch0
- 89_H ADC0809のch1
- 8A_H ADC0809のch2
- 8B_H ADC0809のch3
- 8C_H ADC0809のch4

A/DコンバータにはNS（ナショナルセミコンダクタ）社のADC0809というLSIを使用しました。このLSIは、

- 1) 5V単一電源で動作可能
- 2) 8チャンネルの入力端子を持つ
- 3) TTL, CMOSコンパチ
- 4) CPUと直結可能

などの特徴を持っています。姉妹品としてA/D変換の精度を上げたADC0808（確か3,000円くらいだったかなあ？）というLSIもありますが、今回はそんなに精度は必要

まず、A/Dコンバート開始時にI/Oポー

A/Dコンバートの結果はこれまたI/Oポート88-8F_Hのどこかに対してCPUが読み

ここらへんの処理はLS02(NORゲート)などを使うべきなのですが、ICの節約ということで、余ったLS04 (NOTゲート)とLS32 (ORゲート) を組み合わせて使っています。

The diagram illustrates the hardware configuration for the 8255A-5 PPI chip. The LS138 decoder is used to generate address signals for the 8255A-5. The LS32 gates are used to generate RD and WR signals. The ADC0809 is connected to the 8255A-5 data bus and provides an EOC signal back to the 8255A-5. A 20kΩ resistor is connected to the REF(-) pin of the ADC0809.

The diagram illustrates the pin assignments for the X68000 microprocessor. It is organized into three main sections: Input (入力), Output (出力), and other pins.

Input (入力) Pins:

- PA0-PA7:**
 - PA0: A
 - PA3: B
 - PA2: B'
 - PA1: B'
 - PA0: G (SELECT)
 - PA39: F (START)
 - PA38: E₂
 - PA37: E₁
- PB0-PB7:**
 - PB0: D
 - PB1: C
 - PB20: Bトリガ自動連射
 - PB21: Bトリガ手動
 - PB22: Aトリガ連射
 - PB23: アナログ・デジタル切り替えスイッチ
 - PB24: A/Bボタン切り替えスイッチ
 - PB25: (connected to ground)

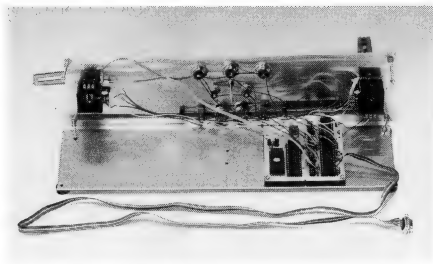
Output (出力) Pins:

- PC0-PC7:**
 - PC14: IOA0
 - PC15: IOA1
 - PC16: IOA2
 - PC17: IOA3
 - PC13: IOA5 (XHL)
 - PC12: IOA6 (XACK0)
 - PC11: (connected to ground)
 - PC10: (connected to ground)
 - PC7: (connected to ground)

Other Pins:

- INT:** Interrupt pin, connected to ground.
- I/O C4:** I/O pin, connected to ground.
- Vcc:** Power supply pin, connected to ground.
- GND:** Ground pin, connected to ground.

A note indicates that the X68000 has 80 pins in total, with the remaining pins being joystick pins.



内部。ボリュームに注目

ちなみに、このあたりの回路は参考文献1を参考にさせていただきました。

PPI (8255A-5のこと) は、コントローラワード\$92を指定してあります。つまり、ポートA、Bは入力用に、ポートCは出力用に使用しています。

X68000からの転送要求命令であるIOC4は、PPIにつながずに直接Z80のINTにつないでいます。この割り込み要求は一定の時間ごとにくるので一種のウォッチドッグタイマの役割も兼ねています。

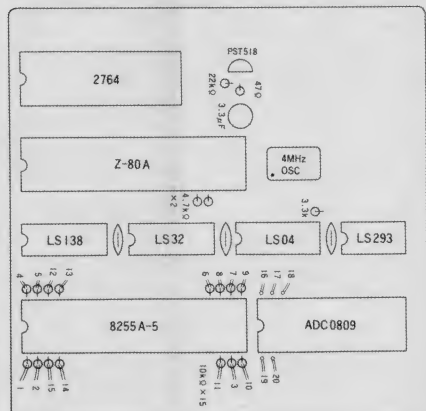
作ってみよう

では、いよいよ製作に取りかかるとしましょう。

まず図1の回路図をコピーしてきます。そして、配線が終わったところから、順に、色鉛筆などで、塗り潰してゆきます。こうすると配線漏れがなくなりますので、初心者の方にはおすすめです。

回路図では、省略されていますが、それぞれのICには、バイパスコンデンサとして、 $0.1\mu\text{F}$ 程度のセラミックコンデンサを電源の+5VとGNDピンのあいだに入れておいてください。このバイパスコンデンサというのは、電源ラインに発生するノイズを低減させる働きをします。基板は、サンハヤトのICB504というのを使いましたが、このまま使うとケースに収まりません

図2 部品配置図



ので、1行のところで切って使います（基板上には、アルファベットと数字が縦横方向に打ってあり、たいへん便利！）。

ICの配置はどのようでもよいのですが、図2のようにするともっとも配線量が少なくて済むと思います。24ピン以上のLSIには必ずソケットを使ってください。この下にパソコンが収納できます。写真ではクロックオシレータにICソケットを用いてますが、これは別になくてもよいと思います。

配線材には、普通のビニール線を用いるとザルソバみたいになり盛上がりしてしまいますので、ラッピング用の線材を用いると良いと思います。

ケースは、これまた各自で好きなように作ってください。私は、写真のようにアルミ板2枚を組み合わせて作りました。最初真ん中のネジはなかったのですが、ABCボタンを押すたびに、ケースがペコペコするのであとからつけておきました。今回のような製作にぴったりのケースが見当たらなかったための、苦肉の策ですが、結構安く上がったので気に入ってます。

芯ケーブルは各線ごとに色がついているのがよいでしょう。私はGND線は黒く

きや絶対にヤダという人間なので、白い線を外して9芯ケーブルを作りました。コネクタの配線には、十分注意してください。ここがX68000を壊すかもしれない唯一の場所です（ただ、X68000は+5VとGNDをショートさせてもシステムリセットをかけ続けるだけでなんとともならないという噂もある……未確認）。

回路図には、表れていませんが、スイッチやボリュームの位置にも気をつけてください。これを間違えると、スティックを右に倒したつもりなのに、F-14は左に倒してしまったりとか、デジタルモードで動かしているつもりが、アナログモードで動いたりとかいう事態に陥ってしまいます。

スティック部は私は、近くの金物屋に行って、写真のようなL字金具をたくさん買ってきて組み合わせて使いましたが、こういうのはだいたい、必要な部品の3倍くらい用意して、ああでもない、こうでもない、と悩みながら積み木遊びをしていると、いずれ妥協案が出るものです。

セットが完成したら、動作確認を行います。ここで、いきなり、アフターバーナーを走らせてもよいのですが、とりあえず、

部品について

今回の製作は電子回路部については特別な部品はありません。問題は機械部です。セット全体でいくらかかるかというのは、だいたいこの部分で差がつきます。私は写真のようにしましたが、ケース部にリプトン紅茶のブリキケース、スロットル部にシャンプーのタンクの柄の部分、スティック部に折れていらなくなったスキーのストックの柄の部分、などというように、身近な廃品を使えば、かなり安くできると思います（私も写真撮影がなかったらそうしたかもしれない）。

まず、写真のようなボリュームを探してきます。今回の製作はこのパーツがないと始まりません（自作してもよいが、かなりハードになりそう）。私は、千石電機で300円で売っているのを買った直後、秋月電子で参考回路図付きで200

円で売っているのを見つけました。一瞬、まったく同じものかと思いましたが、前者のほうがやや丈夫なようでした。ここは、かなりの力がかかる場所ですので、なるべく丈夫なものがよいでしょう。

お金に糸目をつけないければ（このセットの製作意図と矛盾しますが）、電動車椅子取り付け用のポテンショメーターだとか、産業ロボット操作用のスティックなんかあればベターです。もし、中古のパワーショベルとか、米空軍払い下げのF-15のコックピットなんかあればベストです。もし、これらの部品が手に入れば、本家本元のサイバースティックよりも丈夫なものができます（入手に成功した方はぜひ編集部まで一報ください）。

表1：部品表

抵抗	74LS138	55
47Ω	74LS293	65
3.3k	Z80A	210
4.7k	8255A	180
10k	2764-20	800
22k	ADC0809	900
可変抵抗	リセット素子 PST518	50
20k (B型)	クロックオシレータ	4,000MHz 800
20k (B型)	基板	
なもの)	サンハヤト ICB504	210
コンデンサ	スイッチ類	
電解 3.3μF	マイクロスイッチ	× 5
その他パスコンとして0.1μFくらいの	押しボタンスイッチ	× 5
セラミックコンデンサを8個	トグルスイッチ (2P)	× 4
IC	トグルスイッチ (3P)	× 1
74LS04	アルミ板 2t × 400 × 100	× 2
74LS32	ゴム足	× 5

1990年5月号の栗野氏のプログラムを使って確認しましょう。

このとき、アナログ/デジタル切り替えスイッチはアナログ側にしておかないとテストが行えませんので注意してください(私だけが、こんな点にうっかりするのかなあ?)。

ここで、A/B切り替えスイッチや、連射スイッチの動作を確かめ、ボタンの付け間違いがないかを確認しておいてください。

アナログモードで動けば、ROMの書き間違いでもない限りデジタルモードは動作するのですが、一応確認するというか、完成の喜びに浸るという意味でグラディウスでも遊んでみましょう。

ここまできたら、あとはもう、アフターバーナーでもサンダーブレードでも好きなだけ遊んでみましょう。

ソフトについて

それではいよいよ制御用ソフトの説明に入りたいと思います。

最初、このセットはPC-8001mkIIを使って組んでありました。そして、制御プログラムの開発はここから行いました。PC上では、Z80は原則的にモード2で動作する、とか、DMAが一定周期でかかってくるので動作速度が違う、だとかいろいろありましたけれど、結局、組み込みコンピュータの開発としてはこのスタイルが一番なのではないかと思います。そりゃ、ICEがあれば、BESTだけど、あれって1台安くても20万円ぐらいするんですよ。

先ほどから何度も述べましたように、今回のセットには経費削減のため、じゃなくてコンパクト化のためRAMを積んでいません。すべての情報はZ80がレジスタに覚えています。その内訳はだいたい以下のとおりです。

B: Aボタン連射時のカウンタ

C: Bボタン連射時のカウンタ

D上位: ABボタンのデータ(1回目に送る分)

D下位: ABボタンのデータ(7回目に送る分)

E: 右側スティックの前後角度

H: 右側スティックの左右角度

L: 左側スティックの前後角度

リストを見ただければわかると思いますが、大変似かよったところが多々見られます。これはRAMを持たないため、サブルーチンコールをできないばかりでなく、ワークエリアを一切持てないので、「PC

(Programming Counter)が「いまだここにいるか」、ということ自体にある種の情報を持たせているのです。具体的には、PCがラベル「angR……」というところに行れば、もうそれでアナログモードでA/Bボタン判定スイッチが有効(Reverse Mode)ということの意味します。

アナログモードにおいて、X68000から転送要求があると、Z80はそのとき行っている仕事を止めて無条件に0038_H番地へ飛びます。するとここには、

JP SEND

という命令が書かれていますので、データ転送ルーチンSEND:へ飛びます。本来なら、SEND:での仕事が終わると、CPUは元の仕事に戻るべきですが、今回のセットではRAMを積んでいないので戻るべきアドレス(元の仕事のありか)を記憶しておけません。

そこで今回は強引にSEND:での仕事が終わるとCPUは必ずプログラムの一番初めに戻ることにしました。つまりX68000からの転送要求命令はZ80のリセット命令の役割も兼ねているのです。

今回のセットではデータの転送速度は固定です。わざわざ、組み込み型コンピュータを入れておいてそれは手抜きじゃないの、と思うかもしれませんが、そうです。手抜きです。とりあえずのところ、X68000やPC-9801がアナログスティックの動作に追いつかないなんてことはないと思いますので安心してください。

さらにばらしてしまおうと、転送スピードもあまり厳密ではありません。これも手抜きです。結構ラフにデータを垂れ流しても、

ちゃんとX68000が拾ってくれるような転送方式ですので心配はないと思います。

0000_H番地付近と0038_H番地付近に、

JP 0FFFFH

というのが並んでいます。これはROMを何回か有効に利用しようとするものです。ROMの書き込みとは、ROMの初期値である0FF_Hに対して、どこぞのビットを0にしていく、つまり、書き込みたい値と論理和を取っていくということです。

いったんバグの抜け切らないプログラムをROMに書き込んでしまっても、次のプログラムでは、その領域を、

00 (=NOP)

で埋めておけば、ROMを消去することなしに使用することができます(本当はROMっぽく動くRAMで作ったROMエミュレータというのを持っていたんだけど、壊れちゃったんだよう……)。

2回目以降はリセットされてくる0000_H番地と割り込みがかかってくるとコールされる0038_H番地は00_Hで埋めて、代わりに、その次の0003_H番地と003B_H番地のジャンプ命令を有効にすればよいのです。

後日談

「サイバースティックの23,800円は許せない。たかだかジョイスティックだろうが!」というのが、今回の製作の動機でした。しかし、実際に作ってみて「いやあ〜23,800円というのは、なかなか良心的じゃあないかい?」というのが感想です。

電子回路の部分は、まあまあ満足のいく値段に抑えることができたのですが、機械

A/Dコンバータについて

アナログな情報をコンピュータで扱えるようなデジタル情報に変換するインタフェースのことをA/Dコンバータ(Analog Digital Converter)と呼びます。

今回の製作ではジョイスティックの傾き具合をポリウムを用いてアナログ量で表し、それをZ80CPUで処理しています。

A/Dコンバータには大きく分けて積分型のものと逐次比較型のものと、2通りあります。

積分型とは、まず一定時間、測りたい電圧を積分用のコンデンサに加えます。次にそれとは逆の方向に基準電圧を加えていきます。そして、積分用のコンデンサの電荷がなくなるまでの時間をカウントしてやります。この方法は、精度が高く取れるのですが、変換に時間がかかったり、変換用のLSIが高かったりするので今回は見送りました。

もう一方の逐次比較型とは、まず自分で比較電圧を作成し、その電圧と入力電圧の大きさをアナログコンパレータ(比較器)を用いてどち

らが大きいかを判断し、状況に応じて、また比較電圧を作成し比較をしていくというものです。具体的にいうと、まず、基準電圧の1/2と入力電圧を比較します。もし、入力電圧が大きければ最上位ビット(MSB)は1で、小さければ0です。次にその電圧の差ΔV1を基準電圧の1/4と比較します。もし、ΔV1のほうが大きければ、次のビットが1になり、小さければ0になります。そして、その差ΔV2と基準電圧の1/8を比較して……、というように入力電圧の大きさを決定します。

今回使用したADC0809は8ビットの分解能(2の8乗=256段階)を持っていますので、この比較を8回行います。

あ、そうそう、可変抵抗器全般を指してポリウムというのは日本独自の英語だそうで、本当の正しい(?)英語では、Variable Resistorとか、Potentiometerとかいうのだそうです。Volumeだったら略してVLですけど、回路図なんかではポリウムのことをVRと書きますよね。

部分が思ったより高くなってしまいました。それで高いお金をかけた分、よいものが出た上がったかという、そうでもないようです。私のセットではアフターバーナーを1周すると、右手の親指が痛くなってしまいます。これも各自の工夫次第。サイバースティックに飽き足らない人、金はないがハードはまかしとけという人は参考にしてください（参考になるかな……）。

参考文献：

- 1) 作りながら学ぶマイコン設計トレーニング (C Q出版) 神崎 康弘
- 2) Oh!X 1990年5月号 ラジコンスティックの製作 桑野 雅彦

ウォッチドッグタイマ

コンピュータでなにかものを制御する場合、コンピュータの暴走に対しては、かなり気をつけなければなりません。パソコンなどの場合でしただいたい暴走したなあ、とかわかるのですが組み込みコンピュータの場合だとそれを外部から知る術がありません。

1 回暴走したら、それこそ電源を切ってやるまで人知れず黙々と暴走を続けるのです。それなら、いっそのこと電源を切らないまでも、一定時間ごとにシステムにリセットをかけてやれば、たとえ暴走してもその暴走している時間は限られてきます。そのリセットをかけてやるのが、ウォッチドッグタイマ (Watch Dog: 番犬) 回路なのです。

今回のアナログスティックの場合、特別なウォッチドッグタイマ回路は搭載していませんがX68000からの転送要求命令をこれに代用しています。どういことかという、普通は割り込み処理ルーチンでの仕事が終わったら、いまままでやっていた仕事の続きをするものなのですが、今回のアナログスティックはRAMを積んでいません。すなわち、いままでの仕事に戻ろうにも戻るべき仕事を記憶してられないのです。そこで割り込み処理ルーチンから抜ける際には常にシステムの最初の状態、すなわち電源投入時と同じリセットがかかるようにしてやるのです。転んでも、決してタダでは起きない設計だと自分では思っています。

リスト

```

0000 1 :*****
0000 2 :
0000 3 :Cyber Stick Control Program
0000 4 :Programmed By T.Ishigami
0000 5 :'90 Jul 31th
0000 6 :
0000 7 :*****
0000 8 :
0000 P 9 PORTA EQU 00H
0001 P 10 PORTB EQU 01H
0002 P 11 PORTC EQU 02H
0003 P 12 CW EQU 03H
0004 13
0005 P 14 CH0 EQU 08H
0006 P 15 CH1 EQU 09H
0007 P 16 CH2 EQU 0AH
0008 P 17 CH3 EQU 0BH
0009 P 18 CH4 EQU 0CH
0010 19
0011 20 OFFSET 3000H
0012 21 ORG 0000H
0013 22
0014 23 DS 16H ;Masking Pattern
0015 24
0016 C3 44 00
0017 25 JP START
0018 26
0019 C3 FF FF
0020 27 JP 0FFFFH
0021 C3 FF FF
0022 28 JP 0FFFFH ;Reserved For
0023 C3 FF FF
0024 29 JP 0FFFFH
0025 30
0026 31 ORG 003BH
0027 32
0028 C3 E0 02
0029 33 JP SEND
0030 34
0031 C3 FF FF
0032 35 JP 0FFFFH
0033 C3 FF FF
0034 36 JP 0FFFFH
0035 C3 FF FF
0036 37 JP 0FFFFH ;Reserved For
0037 38
0038 39
0039 40
0040 41 START: IN 1 ;Interrupt Mod
0041 42
0042 43 LD A,92H
0043 44 OUT (CW),A
0044 45
0045 46 digital:
0046 47 DI
0047 48
0048 DB 81
0049 49 IN A,(PORTB)
0050 50 BIT 5,A
0051 51 JP Z,analog
0052 52
0053 53 ; Digital Mode
0054 54
0055 55 digA: OUT (CH3),A ;CH3ヲ「アナログ」
0056 56
0057 57 IN A,(PORTA)
0058 58 BIT 0,A
0059 59 JR Z,digA1 ;if A = ON
0060 60 BIT 2,A
0061 61 JR Z,digA1 ;if A' = ON
0062 62 LD H,0
0063 63 LD B,0FFH ;「リセット」信号
0064 64
0065 65 JR digB
0066 66
0067 67 digA1: IN A,(PORTB)
0068 68 BIT 4,A
0069 69 JR Z,digA4
0070 70
0071 71 digA3: IN A,(PORTB) ;「リセット」信号
0072 72 BIT 7,A
0073 73 JR Z,digA3 ;Wait EOC =
0074 74
0075 75 IN A,(CH3)
0076 76 SRL A
0077 77 JR A,CH3
0078 78 SRL A
0079 79 INC A ;A = (CH3) /
0080 80
0081 81
0082 82
0083 83
0084 84
0085 85 LD A,H
0086 86 XOR 1,A
0087 87 LD H,A
0088 88 JR digB
0089 89
0090 90 digA4: LD H,1 ;Normal
0091 91
0092 92 digB: OUT (CH4),A ;CH4ヲ「アナログ」
0093 93
0094 94
0095 95
0096 96
0097 97
0098 98
0099 99
0100 00
0101 01
0102 02
0103 03
0104 04
0105 05
0106 06
0107 07
0108 08
0109 09
0110 10
0111 11
0112 12
0113 13
0114 14
0115 15
0116 16
0117 17
0118 18
0119 19
0120 20
0121 21
0122 22
0123 23
0124 24
0125 25
0126 26
0127 27
0128 28
0129 29
0130 30
0131 31
0132 32
0133 33
0134 34
0135 35
0136 36
0137 37
0138 38
0139 39
0140 40
0141 41
0142 42
0143 43
0144 44
0145 45
0146 46
0147 47
0148 48
0149 49
0150 50
0151 51
0152 52
0153 53
0154 54
0155 55
0156 56
0157 57
0158 58
0159 59
0160 60
0161 61
0162 62
0163 63
0164 64
0165 65
0166 66
0167 67
0168 68
0169 69
0170 70
0171 71
0172 72
0173 73
0174 74
0175 75
0176 76
0177 77
0178 78
0179 79
0180 80
0181 81
0182 82
0183 83
0184 84
0185 85
0186 86
0187 87
0188 88
0189 89
0190 90
0191 91
0192 92
0193 93
0194 94
0195 95
0196 96
0197 97
0198 98
0199 99
0200 00
0201 01
0202 02
0203 03
0204 04
0205 05
0206 06
0207 07
0208 08
0209 09
0210 10
0211 11
0212 12
0213 13
0214 14
0215 15
0216 16
0217 17
0218 18
0219 19
0220 20
0221 21
0222 22
0223 23
0224 24
0225 25
0226 26
0227 27
0228 28
0229 29
0230 30
0231 31
0232 32
0233 33
0234 34
0235 35
0236 36
0237 37
0238 38
0239 39
0240 40
0241 41
0242 42
0243 43
0244 44
0245 45
0246 46
0247 47
0248 48
0249 49
0250 50
0251 51
0252 52
0253 53
0254 54
0255 55
0256 56
0257 57
0258 58
0259 59
0260 60
0261 61
0262 62
0263 63
0264 64
0265 65
0266 66
0267 67
0268 68
0269 69
0270 70
0271 71
0272 72
0273 73
0274 74
0275 75
0276 76
0277 77
0278 78
0279 79
0280 80
0281 81
0282 82
0283 83
0284 84
0285 85
0286 86
0287 87
0288 88
0289 89
0290 90
0291 91
0292 92
0293 93
0294 94
0295 95
0296 96
0297 97
0298 98
0299 99
0300 00
0301 01
0302 02
0303 03
0304 04
0305 05
0306 06
0307 07
0308 08
0309 09
0310 10
0311 11
0312 12
0313 13
0314 14
0315 15
0316 16
0317 17
0318 18
0319 19
0320 20
0321 21
0322 22
0323 23
0324 24
0325 25
0326 26
0327 27
0328 28
0329 29
0330 30
0331 31
0332 32
0333 33
0334 34
0335 35
0336 36
0337 37
0338 38
0339 39
0340 40
0341 41
0342 42
0343 43
0344 44
0345 45
0346 46
0347 47
0348 48
0349 49
0350 50
0351 51
0352 52
0353 53
0354 54
0355 55
0356 56
0357 57
0358 58
0359 59
0360 60
0361 61
0362 62
0363 63
0364 64
0365 65
0366 66
0367 67
0368 68
0369 69
0370 70
0371 71
0372 72
0373 73
0374 74
0375 75
0376 76
0377 77
0378 78
0379 79
0380 80
0381 81
0382 82
0383 83
0384 84
0385 85
0386 86
0387 87
0388 88
0389 89
0390 90
0391 91
0392 92
0393 93
0394 94
0395 95
0396 96
0397 97
0398 98
0399 99
0400 00
0401 01
0402 02
0403 03
0404 04
0405 05
0406 06
0407 07
0408 08
0409 09
0410 10
0411 11
0412 12
0413 13
0414 14
0415 15
0416 16
0417 17
0418 18
0419 19
0420 20
0421 21
0422 22
0423 23
0424 24
0425 25
0426 26
0427 27
0428 28
0429 29
0430 30
0431 31
0432 32
0433 33
0434 34
0435 35
0436 36
0437 37
0438 38
0439 39
0440 40
0441 41
0442 42
0443 43
0444 44
0445 45
0446 46
0447 47
0448 48
0449 49
0450 50
0451 51
0452 52
0453 53
0454 54
0455 55
0456 56
0457 57
0458 58
0459 59
0460 60
0461 61
0462 62
0463 63
0464 64
0465 65
0466 66
0467 67
0468 68
0469 69
0470 70
0471 71
0472 72
0473 73
0474 74
0475 75
0476 76
0477 77
0478 78
0479 79
0480 80
0481 81
0482 82
0483 83
0484 84
0485 85
0486 86
0487 87
0488 88
0489 89
0490 90
0491 91
0492 92
0493 93
0494 94
0495 95
0496 96
0497 97
0498 98
0499 99
0500 00
0501 01
0502 02
0503 03
0504 04
0505 05
0506 06
0507 07
0508 08
0509 09
0510 10
0511 11
0512 12
0513 13
0514 14
0515 15
0516 16
0517 17
0518 18
0519 19
0520 20
0521 21
0522 22
0523 23
0524 24
0525 25
0526 26
0527 27
0528 28
0529 29
0530 30
0531 31
0532 32
0533 33
0534 34
0535 35
0536 36
0537 37
0538 38
0539 39
0540 40
0541 41
0542 42
0543 43
0544 44
0545 45
0546 46
0547 47
0548 48
0549 49
0550 50
0551 51
0552 52
0553 53
0554 54
0555 55
0556 56
0557 57
0558 58
0559 59
0560 60
0561 61
0562 62
0563 63
0564 64
0565 65
0566 66
0567 67
0568 68
0569 69
0570 70
0571 71
0572 72
0573 73
0574 74
0575 75
0576 76
0577 77
0578 78
0579 79
0580 80
0581 81
0582 82
0583 83
0584 84
0585 85
0586 86
0587 87
0588 88
0589 89
0590 90
0591 91
0592 92
0593 93
0594 94
0595 95
0596 96
0597 97
0598 98
0599 99
0600 00
0601 01
0602 02
0603 03
0604 04
0605 05
0606 06
0607 07
0608 08
0609 09
0610 10
0611 11
0612 12
0613 13
0614 14
0615 15
0616 16
0617 17
0618 18
0619 19
0620 20
0621 21
0622 22
0623 23
0624 24
0625 25
0626 26
0627 27
0628 28
0629 29
0630 30
0631 31
0632 32
0633 33
0634 34
0635 35
0636 36
0637 37
0638 38
0639 39
0640 40
0641 41
0642 42
0643 43
0644 44
0645 45
0646 46
0647 47
0648 48
0649 49
0650 50
0651 51
0652 52
0653 53
0654 54
0655 55
0656 56
0657 57
0658 58
0659 59
0660 60
0661 61
0662 62
0663 63
0664 64
0665 65
0666 66
0667 67
0668 68
0669 69
0670 70
0671 71
0672 72
0673 73
0674 74
0675 75
0676 76
0677 77
0678 78
0679 79
0680 80
0681 81
0682 82
0683 83
0684 84
0685 85
0686 86
0687 87
0688 88
0689 89
0690 90
0691 91
0692 92
0693 93
0694 94
0695 95
0696 96
0697 97
0698 98
0699 99
0700 00
0701 01
0702 02
0703 03
0704 04
0705 05
0706 06
0707 07
0708 08
0709 09
0710 10
0711 11
0712 12
0713 13
0714 14
0715 15
0716 16
0717 17
0718 18
0719 19
0720 20
0721 21
0722 22
0723 23
0724 24
0725 25
0726 26
0727 27
0728 28
0729 29
0730 30
0731 31
0732 32
0733 33
0734 34
0735 35
0736 36
0737 37
0738 38
0739 39
0740 40
0741 41
0742 42
0743 43
0744 44
0745 45
0746 46
0747 47
0748 48
0749 49
0750 50
0751 51
0752 52
0753 53
0754 54
0755 55
0756 56
0757 57
0758 58
0759 59
0760 60
0761 61
0762 62
0763 63
0764 64
0765 65
0766 66
0767 67
0768 68
0769 69
0770 70
0771 71
0772 72
0773 73
0774 74
0775 75
0776 76
0777 77
0778 78
0779 79
0780 80
0781 81
0782 82
0783 83
0784 84
0785 85
0786 86
0787 87
0788 88
0789 89
0790 90
0791 91
0792 92
0793 93
0794 94
0795 95
0796 96
0797 97
0798 98
0799 99
0800 00
0801 01
0802 02
0803 03
0804 04
0805 05
0806 06
0807 07
0808 08
0809 09
0810 10
0811 11
0812 12
0813 13
0814 14
0815 15
0816 16
0817 17
0818 18
0819 19
0820 20
0821 21
0822 22
0823 23
0824 24
0825 25
0826 26
0827 27
0828 28
0829 29
0830 30
0831 31
0832 32
0833 33
0834 34
0835 35
0836 36
0837 37
0838 38
0839 39
0840 40
0841 41
0842 42
0843 43
0844 44
0845 45
0846 46
0847 47
0848 48
0849 49
0850 50
0851 51
0852 52
0853 53
0854 54
0855 55
0856 56
0857 57
0858 58
0859 59
0860 60
0861 61
0862 62
0863 63
0864 64
0865 65
0866 66
0867 67
0868 68
0869 69
0870 70
0871 71
0872 72
0873 73
0874 74
0875 75
0876 76
0877 77
0878 78
0879 79
0880 80
0881 81
0882 82
0883 83
0884 84
0885 85
0886 86
0887 87
0888 88
0889 89
0890 90
0891 91
0892 92
0893 93
0894 94
0895 95
0896 96
0897 97
0898 98
0899 99
0900 00
0901 01
0902 02
0903 03
0904 04
0905 05
0906 06
0907 07
0908 08
0909 09
0910 10
0911 11
0912 12
0913 13
0914 14
0915 15
0916 16
0917 17
0918 18
0919 19
0920 20
0921 21
0922 22
0923 23
0924 24
0925 25
0926 26
0927 27
0928 28
0929 29
0930 30
0931 31
0932 32
0933 33
0934 34
0935 35
0936 36
0937 37
0938 38
0939 39
0940 40
0941 41
0942 42
0943 43
0944 44
0945 45
0946 46
0947 47
0948 48
0949 49
0950 50
0951 51
0952 52
0953 53
0954 54
0955 55
0956 56
0957 57
0958 58
0959 59
0960 60
0961 61
0962 62
0963 63
0964 64
0965 65
0966 66
0967 67
0968 68
0969 69
0970 70
0971 71
0972 72
0973 73
0974 74
0975 75
0976 76
0977 77
0978 78
0979 79
0980 80
0981 81
0982 82
0983 83
0984 84
0985 85
0986 86
0987 87
0988 88
0989 89
0990 90
0991 91
0992 92
0993 93
0994 94
0995 95
0996 96
0997 97
0998 98
0999 99
1000 00
1001 01
1002 02
1003 03
1004 04
1005 05
1006 06
1007 07
1008 08
1009 09
1010 10
1011 11
1012 12
1013 13
1014 14
1015 15
1016 16
1017 17
1018 18
1019 19
1020 20
1021 21
1022 22
1023 23
1024 24
1025 25
1026 26
1027 27
1028 28
1029 29
1030 30
1031 31
1032 32
1033 33
1034 34
1035 35
1036 36
1037 37
1038 38
1039 39
1040 40
1041 41
1042 42
1043 43
1044 44
1045 45
1046 46
1047 47
1048 48
1049 49
1050 50
1051 51
1052 52
1053 53
1054 54
1055 55
1056 56
1057 57
1058 58
1059 59
1060 60
1061 61
1062 62
1063 63
1064 64
1065 65
1066 66
1067 67
1068 68
1069 69
1070 70
1071 71
1072 72
1073 73
1074 74
1075 75
1076 76
1077 77
1078 78
1079 79
1080 80
1081 81
1082 82
1083 83
1084 84
1085 85
1086 86
1087 87
1088 88
1089 89
1090 90
1091 91
1092 92
1093 93
1094 94
1095 95
1096 96
1097 97
1098 98
1099 99
1100 00
1101 01
1102 02
1103 03
1104 04
1105 05
1106 06
1107 07
1108 08
1109 09
1110 10
1111 11
1112 12
1113 13
1114 14
1115 15
1116 16
1117 17
1118 18
1119 19
1120 20
1121 21
1122 22
1123 23
1124 24
1125 25
1126 26
1127 27
1128 28
1129 29
1130 30
1131 31
1132 32
1133 33
1134 34
1135 35
1136 36
1137 37
1138 38
1139 39
1140 40
1141 41
1142 42
1143 43
1144 44
1145 45
1146 46
1147 47
1148 48
1149 49
1150 50
1151 51
1152 52
1153 53
1154 54
1155 55
1156 56
1157 57
1158 58
1159 59
1160 60
1161 61
1162 62
1163 63
1164 64
1165 65
1166 66
1167 67
1168 68
1169 69
1170 70
1171 71
1172 72
1173 73
1174 74
1175 75
1176 76
1177 77
1178 78
1179 79
1180 80
1181 81
1182 82
1183 83
1184 84
1185 85
1186 86
1187 87
1188 88
1189 89
1190 90
1191 91
1192 92
1193 93
1194 94
1195 95
1196 96
1197 97
1198 98
1199 99
1200 00
1201 01
1202 02
1203 03
1204 04
1205 05
1206 06
1207 07
1208 08
1209 09
1210 10
1211 11
1212 12
1213 13
1214 14
1215 15
1216 16
1217 17
1218 18
1219 19
1220 20
1221 21
1222 22
1223 23
1224 24
1225 25
1226 26
1227 27
1228 28
1229 29
1230 30
1231 31
1232 32
1233 33
1234 34
1235 35
1236 36
1237 37
1238 38
1239 39
1240 40
1241 41
1242 42
1243 43
1244 44
1245 45
1246 46
1247 47
1248 48
1249 49
1250 50
1251 51
1252 52
1253 53
1254 54
1255 55
1256 56
1257 57
1258 58
1259 59
1260 60
1261 61
1262 62
1263 63
1264 64
1265 65
1266 66
1267 67
1268 68
1269 69
1270 70
1271 71
1272 72
1273 73
1274 74
1275 75
1276 76
1277 77
1278 78
1279 79
1280 80
1281 81
1282 82
1283 83
1284 84
1285 85
1286 86
1287 87
1288 88
1289 89
1290 90
1291 91
1292 92
1293 93
1294 94
1295 95
1296 96
1297 97
1298 98
1299 99
1300 00
1301 01
1302 02
1303 03
1304 04
1305 05
1306 06
1307 07
1308 08
1309 09
1310 10
1311 11
1312 12
1313 13
1314 14
1315 15
1316 16
1317 17
1318 18
1319 19
1320 20
1321 21
1322 22
1323 23
1324 24
1325 25
1326 26
1327 27
1328 28
1329 29
1330 30
1331 31
1332 32
1333 33
1334 34
1335 35
1336 36
1337 37
1338 38
1339 39
1340 40
1341 41
1342 42
1343 43
1344 44
1345 45
1346 46
1347 47
1348 48
1349 49
1350 50
1351 51
1352 52
1353 53
1354 54
1355 55
1356 56
1357 57
1358 58
1359 59
1360 60
1361 61
1362 62
1363 63
1364 64
1365 65
1366 66
1367 67
1368 68
1369 69
1370 70
1371 71
1372 72
1373 73
1374 74
1375 75
1376 76
1377 77
1378 78
1379 79
1380 80
1381 81
1382 82
1383 83
1384 84
1385 85
1386 86
1387 87
1388 88
1389 89
1390 90
1391 91
1392 92
1393
```


カード型データベース(2)

Izumi Daisuke 泉 大介

BASIC

X-BASICの総集編として取り上げたカード型データベース。今回はデータ構造の基本設計ができたところですが、データベースというからには検索やソートといった機能が必要ですね。これらの機能をcommand関数として付け加えていきましょう。

先月はカード型データベースのできるだけ簡単な仕様を決め、それをプログラムする過程をお届けしたわけですが、いかがだったでしょうか。データがディスク内にどのように保存されるのかおわかりいただけたでしょうか。

今月はカード型データベースの完成編です。今月まわしにしたcommands関数の説明をし、カードの検索、ソートを付け加えていきます。

メニュー選択ルーチン

先月プログラムを入力して動かしてみた方はおわかりかと思いますが、このカード型データベースは最下行に、

1/3 : Edit, Del, Clear, Quit

というメニューを表示し、その先頭のEDCQの文字をキーボードから入力することで処理が行われます。これはcommands関数の仕事です。ではその内容を見ていくことにしましょう。

プログラムはコンパイルすることを考えて若干の変更が加えてあります。行末に「追加」「変更」のコメントがある行は先月のものと異なっていますので注意してください。また、先月掲載したinputData関数内のline変数は、コンパイル時にエラーとなってしまう¹⁾。変数名をlnに変更してください。line\$変数のほうはそのままでOKです。では先頭から見ていきましょう。

2つの大域変数を追加しました。これらは今月追加した検索、ソート機能で使います。それぞれを説明するときに具体的にどう使っているのかを紹介することにします。

commands関数は4690行から始まります。最初は変数の宣言です。cardpはcards配列内の位置を表すのに使います。usingCardsは現在使用しているカードの総数、chosen、modeは今月追加した変数で、それぞれcards内に収められているカード数、検索モード保持用に使用します。

chosen変数の追加には疑問を持たれた方もいらっしゃるでしょう。これはcards配列の役割に関係があります。先月は使用されているカードがすべてcards配列に収められていましたが、本来これは選択されたカードだけを収める目的で用意した配列です。

検索をかけるとき、検索対象となるカードはcards配列に登録されているものだけです。したがって検索条件をだんだんと厳しくしていけば、cards配列に残るカードはどんどん少なくなっていきます。

ソートするときも同様で、並び替えられるカードはcards配列に登録されているものだけが対象となります。そこで現在cards配列にいくつかのカードが収められているのかを保持する変数を用意したというわけです。

commands関数は最初に画面を32行モードにした後、続いてcollectCards関数を呼び出して使用中のカードをすべてcards配列にセットします。これが初期状態です。collectCards関数が返した使用カード数はusingCardsにセットされ、chosenにもこれが代入されます。最初に表示するのはcards配列の先頭に入っているカードですからcardpは0となり、readCard関数でcards(cardp)のカード、すなわちcards(0)のカードが表示されます。ここまでが初期設定です。

続いてメニューの表示、その選択、機能の実行というループが続くことになります。メニュー表示は若干変更してあり、

0/3 (5) : Edit, ……

という形になっています。最初の数字はcards配列中のどこにいるのかを表しています。次の数字はcards配列に登録されているカードの総数です。そして最後のカッコ内の数字は使用中のカードの総数を意味しています。メニューに表示される機能は以下のとおりです。

Edit	カードの修正
Del	表示中のカードを削除
Clear	cards配列の初期化
eXclude	表示中のカードをcards配列から外す
Find	カードの検索
Sort	カードのソート
Quit	データベースの終了

いずれも大文字になっている英字を入力すれば動作します。

これをinkey\$で入力してもらい、switchで処理を分ければメニュールーチンは終了です。簡単ですね。ではメニューごとに分けて見ていくことにしましょう。

1) これはX-BASICのグラフィック関数lineと同じ名前であるためです。X-BASICでは動くので、うっかりそのままにしてみました。

各機能の解説

●Edit：カードのエディット

カードのエディットはinputData関数で行います。まずカーソルを(0, 0)に移動していますが、これはコンパイル後の動作がX-BASICと異なっていたためです。カーソルが32行目にあるときに画面を31行モードにすると、画面が1行スクロールアップしてしまうのです。画面モードを変更しデータの入力が終わったら、再び画面を32行モードに戻します。ここで、エディットしたカードがすでに使用されているカードかどうかをチェックします。未使用のカードなら(新しいデータを入力したのなら)newCard関数によって新しいカードをひとつもらい、カード数を変更。最後にwriteCard関数でカードをディスクにセーブします。

●ROLL UP, ROLL DOWN：カードのスクロール

これはcardpを増減し、それに応じたカードをディスクから読み出して画面に表示するだけです。先月はusingCardsでROLL UP時のチェックを行っていましたが、ここはchosen以上にスクロールできないよう変更を加えてあります。

●Del：表示中のカードを削除

先月、カードのディスク上での保存方法を説明しましたが、カードの削除はこれと密接に関係していますので処理が少々面倒です。dbasep配列は実際にデータが収められている位置を示していますから、まずはこれをクリアしなければなりません。対応するusedRecsに0を入れると同時に、dbasepの該当位置を-1にしてデータをクリアします²⁾。

続いて全部-1となったdbasepを～.ratファイルに保存します。これでこのカードはデータをまったく持たない新しいカードとなりました。usedCardsの該当位置のほうも0にして、カードを未使用状態に戻します。最後にusedRecsとusedCardsをratファイルに書き出せば削除作業は終了です。

残る仕事は削除したカードをcards配列から外すことです。cardpから最後のカードまでを順に繰り上げていき、削除したカードをcards配列から消します。そして次のカードを表示、usingCards、chosenを更新します。

●Clear：cards配列の初期化

先ほども説明したように、cards配列は「選択されたカード」を収めます。検索を繰り返していくうちにcards配列に残るカードはどんどん少なくなっていくことでしょう。これを初期化し、使用されているカードをすべてcards配列に収め直すのがこのコマンドです。commands関数の先頭でやったのと同じように、collectCards関数を呼び出して処理します。

●Quit：データベースの終了

コマンド入力ループはflag変数が1である間回り続けます。flag変数に0を代入すればwhile～end

whileループは終了となります。

●eXclude：表示中のカードをcards配列から外す

ここからが、今月新たに追加したコマンド群です。検索、ソートの対象となるcards配列中のカードですが、なかには明らかに意に沿わないため条件を満たしてはいるが対象から外そうというものも出てきます。2コンの重さはどうにも我慢できないからヤダとか、ペン太ックスはホールディングがいまいち好みではないから外そうとか。Xコマンドはこの処理を行います。

プログラムはDelコマンドを簡略化した形になっています。cards配列から現在表示しているcardp番目のカードを取り除き、残りのカードを詰めるだけです。

●Find：カードの検索

さあ、本日の目玉商品が登場しました。ここまで何度もいつてきたように、これはcards配列に登録されているカードから、条件に合うカードを見つけ出してcards配列に再セットしようというコマンドです。でも、本当にcards配列の中から探すだけでいいのでしょうか。「重量が600g未満で幅が155mm未満のカメラ!」と指定して探した結果に満足できず、「では重量が600g未満ならよしとしよう」と条件を緩めた検索結果を加えたい、あるいは「多重露出ができれば重量も幅もどうでもいい」と思い切り譲歩して検索した結果も合わせて見たいという要求はないのでしょうか。

これを解決するため、検索に2つのモードを設けることにしました。ひとつはノーマル検索で、これはcards配列中のカードから条件に合致するものを探します。もうひとつは追加検索でこちらはcards配列に登録されたカード以外のものが検索対象となります。Fキーが押されると画面最下行はこの検索モードの入力となります。

Normal, Additional >

と表示し、nかa、あるいはESCキーが押されるまでループ。キー入力を変数modeに格納します。ESCキーが押された場合は検索中断ということで処理終了です。

続いて検索条件の入力です。これはgetKey関数で行います。getKey関数を見てください。条件の入力はinputData関数を使います。カードへのデータ入力やカードの設計に使っているアレですね。入力されたデータをData配列に収めるのがinputData関数の役目でした。

プログラムはまずData配列をクリアし、条件入力のループに入ります。画面をクリアしたあとData配列になにか入っていたらその行を画面の該当位置に表示するのですが、最初は当然なにも入っていませんので表示なしです。31行モードにしてinputData関数を呼び、条件を入力してもらいます。そう、inputData関数で入力するので、カードにデータを入れるのと同じ方法で条件も指定するのです。具体的に説明しましょう。

2) 実際にdatファイルに収められている文字データのほうをクリアする必要はありません。どのカードからも参照されなくなったということはクリアしたというのと同じですからね。

3) 重量600g未満で幅155mm未満のカメラ、というのはANDで検索することになります。

4) 重量600g未満か多重露出ができるカメラ、というのはOR検索です。

重量 #
と設計されている場所があれば、データ入力時に、
重量 ■
とカーソルが#の場所で点滅し、カメラの重量を入力できるのは先月説明したとおりです。同じinput Data関数を使っているのですから、条件の入力時にもカーソルはここで点滅します。もし600gより軽いカメラを探したいのなら、ここで、

重量 <600
と入力します。これは「重量が600より小さい」と読めますね。また、多重露出のできるカメラを探したいなら、

多重露出 =○
とでもなるでしょう。もちろんデータが、
多重露出 ○
あるいは、
多重露出 ×
という形で有無を区別している場合の話ですが。
このように検索条件は、

「演算子」+「比較データ」
という形で与えます。使える演算子は、
= 比較データと等しい
< 比較データより小さい
> 比較データより大きい
! 比較データと等しくない
) 比較データが含まれている
(比較データに含まれている
の6つです。>=などと組み合わせて使うことはできません。悪しからず。

検索条件の入力が終了したら、Data配列を調べて条件がちゃんと与えられているかどうかを調べます。Data配列の中でデータの入っているものは、先頭が必ずこれら6つの演算子でなければなりませんのでそれをチェックするのです。間違いがあればその行をシアンで表示し、入力のやり直しです。

では、commands関数に戻りましょう。getKey関数で検索条件の入力が終了したら検索の開始……、いえいえ、検索にはもうひとつ考えておかねばならないことがあります。それは、列挙した複数の

条件をすべて満たすものを探す(ANDで検索する³⁾)のか、ひとつでも満たしていればOKとする(ORで検索する⁴⁾)のかです。そこで最後に検索方法の入力メニューを表示します。これも画面最下行に、

検索方法: And, Or >

と表示して入力してもらいます。ここでもESCで中断できるようにしておくのがいいでしょう。入力してもらった検索方法は、最初に出たconditionという大域変数に0か1を代入して保存しておきます。検索を行う関数はこの変数を参照し、検索方法を決定します。

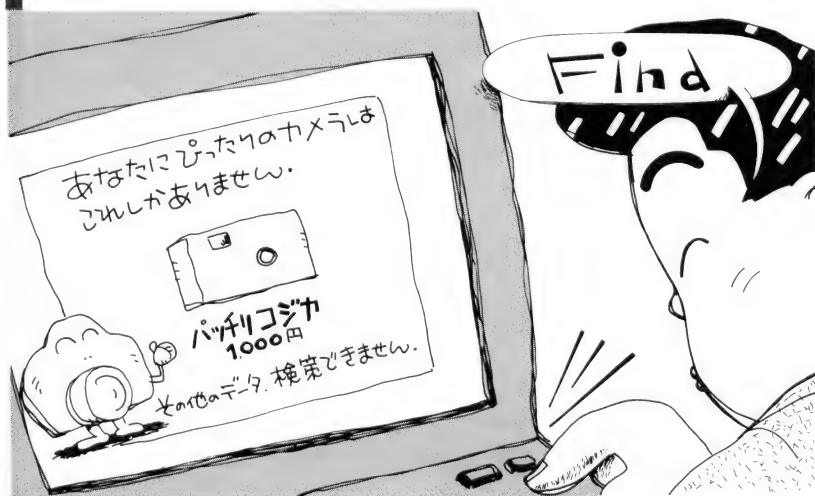
前準備が整ったら検索開始です。ノーマル検索の場合はinCards関数で、追加検索の場合はaddCards関数で検索を行います。いずれの関数も検索終了時にcards配列に抽出されたカードの総数を返しますので、これをchosen変数に代入してやります。cardpのほうはノーマル検索時にはcards配列の先頭に、追加検索時には追加されたカードの先頭になります。

ではinCards関数です。cards配列内のカードを順にfindCards関数に渡し、そのカードが検索条件に合致するかどうかを判定します。合致すればOKですが、合致しなければそのカードはcards配列から削除されます。

addCards関数のほうはcards配列の代わりにusedCardsから使用されているカードを探し出しfindCards関数に渡します。ただしcards配列に登録されているものは判定対象とはしないというのが仕様ですから、ちょっとした細工が必要です。usedCards配列はカードの使用状況を1と0で区別しています。usedCards(n)=1ならn番目のカードは使われているという印です。ここでcards配列に登録されているカードに対応する数値を2に置き換えてやれば、usedCards配列中で1になっているカードは「使用されていてcards配列には入っていない」カードとなりますね。ああ、簡単。これをfindCards関数に渡してチェックしてもらってください。条件に合致すればそのカードをcards配列に加えていきます。最後にusedCards配列の値を1に戻して終了です。

カード検索のトリを務めるのはfindCards関数です。再びratファイルやdatファイルが登場しますので注意してください。findCards関数は引数として条件と比較するカードの番号を受け取ります。まず最初にやることは、渡されたカードのデータ情報をratファイルからdbasep配列に取り出すことです。続いてData配列を調べ、条件が設定されている行があれば演算子をop変数に、比較データをcond変数にセットし、対応する行のデータをdatファイルからline\$変数に取り出して吟味を始めます。リストでは7180行からが演算子によって処理を分けているところですが、すべて文字列の比較で処理している点に注意してください。

こうしてData配列中の条件を順に調べていくのですが、検索方法によって検索打ち切りの条件が異



なります。AND検索 (condition=1) なら設定されている条件がすべて満たされなければなりませんから、不成立の条件があればそこで打ち切りです。逆にOR検索 (condition=0) なら、ひとつでも条件が成立すればそこで打ち切っていいわけです。条件成立かどうかを返してfindCards関数は終了です。

Sort: カードのソート

commands関数もいよいよ最後の項目になりました。カードのソートです。ここでもgetKey関数を使ってソート条件を入力します。こちらの条件は実に単純です。カメラの重量を昇順にソートしたければ、
重量 <
と書くだけです。逆に降順にソートしたければ'>'を使います。

```
card m < card n < card o
```

という意味だと考えればわかりやすいでしょう。プログラムはgetKey関数で入力されたData配列を調べ、条件が設定されている行を見つけたところでソートを開始します。最初に見つけた場所だけが有効で、複数キーによるソートには対応していません。

実際にソートを行うのはshellSort関数です。その名のとおりシェルソートというアルゴリズムを使ってソートを実行します。ではshellSort関数を見てみましょう。shellSort関数はcards配列中のデータ数、ソートのキーとなる行、そして昇順・降順の区別を引数として受け取ります。まず最初にやることは、cards配列に登録されているカードのソートキーとなる行をdatファイルから取り出し今月追加したkeys配列にセットすることです。キーとなる行は何度も参照されます。そのたびにファイルから読み込んでいたのでは時間がかかり過ぎるだろうと考えて最初に読み込んでおくことにしました。

キーの読み込みが終わったらいよいよシェルソートの開始です。シェルソートはちょっと変わったソ

ートの仕方をします。適当な大きさのgapを考え、i番目のデータとi+gap番目のデータを比較して、大小関係が逆なら入れ替えを行うのです。最初gapはデータ数の半分に設定され、データ列の前半と後半で入れ替えが行われます。次にgapはさらに半分にされ、半分にされ……、ついにgapが0になったときにはデータは綺麗に整列しているという方法です。データのバラツキをまずはザッとならし、次第に細かく整えていくというイメージです⁵⁾。

まあ、今回はソートの特集ではないので深入りするのは止めましょう。shellSort関数では、データの入れ替えが必要となった時点でkeys配列のキーと、それに対応する cards 配列のカード番号を同時に入れ替えています。

さて、来月の調理実習は～

X-BASIC総集編のカード型データベースはいかがでしたか？ なに、inputData関数が遅すぎる？ ごもっとも。本文でも触れているように、このプログラムはXCでコンパイルすることができます。速度に不満のある方はコンパイルして利用してみてください。コンパイルは簡単です。XCのシステムディスクを起動して本プログラムの入ったディスクをBドライブにセット。

b:

で画面に「B>」と表示されたら、

```
cc database.bas
```

とするだけであとは自動的に終了します。実行は、

```
database
```

と入力すればOK。ぜひ挑戦してみてください⁶⁾。

さて、来月は本プログラムをより便利に使うためのツールをいくつか用意し、最後を締めくくりたいと思います。ではまた、来月。

5) これではなんの説明にもなっていませんね。バブルソートなどと比較するとわかりやすいと思うのですが……、なんせページ数がないもので。ごめんなさい、皆さんの努力に期待します。

6) XC ver.2.0でコンパイルすると、山のようにWarningが出ますが、無視して差し支えありません。厳しくなったチェックにBCが対応していないのが原因です。

リスト1 カード型データベース(その2)

```
150 int condition          /* 検索方法
160 str keys[100][97]      /* ソート用キー配列
4660 /*
4670 /* メインメニュー&カード閲覧
4680 /*
4690 func commands()
4700 str ch
4710 int flag = 1
4720 int cardp, usingCards
4730 int chosen
4740 str mode
4750 int i
4760 /*
4770 console 0,32,0          /* 画面を32行モードに変更
4780 usingCards = collectCards() /* 使用中のカード数を数え
4790 chosen = usingCards     /* 「追加」chosenにセット
4800 cardp = 0              /* 最初のカードを
4810 readCard( cards( cardp )) /* 表示する
4820 while flag
4830 locate 0, 31           /* 最下段にメニューを表示
4840 print cardp;"/";chosen;"/";usingCards;"/"; /* 「変更」
4850 print " : Edit, Del, Clear, "; /* 「変更」
4860 print "xclude, Find, Sort, Quit >"; /* 「追加」
4870 ch = inkeys
4880 switch asc( ch )
4890 case 'e'                /* カードのエディット
4900 case 'E'
4910 locate 0, 0
4920 console 0,31,1
4930 inputData( 1 )
4940 console 0,32,0
4950 if cards( cardp ) = 255 then {
4960 cards( cardp ) = newCard()
4970 if usingCards < 99 then { /* 「変更」
4980 usingCards = usingCards + 1
4990 chosen = chosen + 1
5000 }
5010 }
5020 writeCard( cards( cardp )) /* カードのセーブ
5030 case 14                /* ROLL UP
5040 if cardp = chosen then break /* 「変更」
```

```
5050 cardp = cardp + 1      /* 次のカードを
5060 readCard( cards( cardp )) /* 読み込む
5070 break
5080 case 15                 /* ROLL DOWN
5090 if cardp = 0 then break
5100 cardp = cardp - 1
5110 readCard( cards( cardp )) /* 前のカードを
5120 break
5130 case 'd'               /* 表示中のカード削除
5140 case 'D'
5150 if cards( cardp ) = 255 then break
5160 for i=0 to 30          /* レコード解放
5170 if dbasep( i ) <> -1 then {
5180 usedRecs( dbasep( i )) = 0
5190 dbasep( i ) = -1
5200 }
5210 next
5220 fseek( rat, 2100 + cards( cardp ) * 4 * 31, 0 )
5230 fwrite( dbasep, 31, rat ) /* ディスク上のdbasep更新
5240 /*
5250 usedCards( cards( cardp )) = 0 /* カード解放
5260 fseek( rat, 0, 0 )
5270 fwrite( usedCards, 100, rat ) /* カードと
5280 fwrite( usedRecs, 2000, rat ) /* レコードの更新
5290 for i=cardp to 98      /* カードを削除
5300 cards( i ) = cards( i+1 )
5310 next
5320 cards( 99 ) = 255
5330 readCard( cards( cardp )) /* 次のカードを読み込む
5340 usingCards = usingCards - 1
5350 chosen = chosen - 1
5360 break
5370 case 'c'               /* 使用中カードの再整列
5380 case 'C'
5390 usingCards = collectCards()
5400 chosen = usingCards
5410 cardp = 0
5420 readCard( cards( cardp ))
5430 break
5440 case 'q'               /* データベースの終了
5450 case 'Q'
```



```

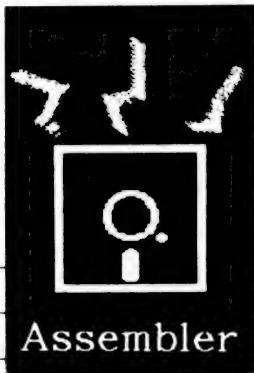
5460 flag = 0
5470 break
5480 /*
5490 /* 以下追加コマンド
5500 /*
5510 case 'x'
5520 case 'X'
5530 if cards( cardp ) = 255 then break
5540 for i=cardp to 98 /* カードをcardsから削除
5550 cards( i ) = cards( i+1 )
5560 next
5570 cards( 99 ) = 255 /* 最後には未使用フラグを
5580 chosen = chosen - 1
5590 readCard( cards( cardp ) ) /* 次のカードを読み込む
5600 break
5610 case 'f' /* カードの検索
5620 case 'F'
5630 repeat /* 検索モード入力
5640 locate 0, 31
5650 print chr$(5); "Normal, Additional > ";
5660 mode = inkey$
5670 until instr( 1, "nNaO" + chr$(27), mode )
5680 if mode = chr$( 27 ) then break
5690 getKey() /* 検索条件入力
5700 repeat
5710 locate 0, 31
5720 print chr$(5); "検索方法: And, Or > ";
5730 ch = inkey$
5740 until instr( 1, "aAoO" + chr$(27), ch )
5750 if ch = chr$(27) then break
5760 if toupper( asc( ch ) ) = 'A' then {
5770 condition = 0 /* 検索方法を保存
5780 } else { /* inCards, addCards関数で
5790 condition = 1 /* 参照される
5800 }
5810 switch asc( mode )
5820 case 'n' /* 通常検索
5830 case 'N'
5840 chosen = inCards( chosen ) /* 合致カード数を更新
5850 cardp = 0 /* cardpは先頭へ
5860 break
5870 case 'a' /* 追加検索
5880 case 'A'
5890 cardp = chosen /* cardpは追加部へ
5900 chosen = addCards( chosen ) /* 合致カード数を更新
5910 break
5920 default
5930 break
5940 endswitch
5950 readCard( cards( cardp ) ) /* カードを表示
5960 break
5970 case 's' /* カードのソート
5980 case 'S'
5990 getKey() /* ソート条件入力
6000 for i=0 to 30
6010 if Data( i ) <> "" then { /* 条件があればソート
6020 shellSort( chosen-1, i, left$( Data( i ), 1 ) )
6030 break
6040 }
6050 next
6060 cardp = 0 /* cardpは先頭へ
6070 readCard( cards( 0 ) ) /* カードを表示
6080 break
6090 default
6100 break
6110 endswitch
6120 endwhile
6130 locate 0, 0 /* 「追加」一旦カーソルを移動
6140 console 0,31,1 /* 31行モードに
6150 endfunc
6160 /*
6170 /* 検索用キー入力
6180 /*
6190 func getKey()
6200 int i, flag
6210 str op
6220 for i=0 to 30
6230 Data( i ) = "" /* Dataをクリア
6240 next
6250 repeat
6260 cls
6270 for i=0 to 30
6280 if Data( i ) <> "" then { /* データがあれば表示
6290 locate dataEntry( i ), i
6300 print Data( i );
6310 }
6320 next
6330 locate 0, 0 /* 一旦カーソルを移動
6340 console 0, 31, 1 /* 31行モードに
6350 inputData( 1 ) /* データを入力
6360 console 0, 32, 0 /* 32行に戻す
6370 /*
6380 cls
6390 flag = 0
6400 for i=0 to 30 /* 入力キーのチェック
6410 if Data( i ) = "" then continue
6420 op = left$( Data( i ), 1 )
6430 if instr( 1, "<>!", op ) then {
6440 color 3 /* 条件がOKなら白で
6450 } else {
6460 flag = 1
6470 color 1 /* さもないとシアンで
6480 }
6490 locate dataEntry( i ), i
6500 print Data( i ); /* データを表示する
6510 next
6520 if flag then { /* 間違いがあれば
6530 color 1 /* シアンで指摘
6540 locate 0, 31
6550 print "条件指定に間違いがあります ";
6560 op = inkey$
6570 }
6580 color 3
6590 until flag = 0 /* 間違いがなくなるまで
6600 endfunc
6610 /*
6620 /* cardsの中から条件に合致するものを調べる
6630 /*
6640 func int inCards( maxNum )
6650 int i, j
6660 i = 0
6670 while i < maxNum /* cards内を順に検索

```

```

6680 if findCards( cards( i ) ) = 0 then { /* 合致しなければ
6690 for j=i to maxNum-1 /* そのデータを削除
6700 cards( j ) = cards( j+1 )
6710 next
6720 cards( maxNum ) = 255 /* 最後のカードを未使用にし
6730 maxNum = maxNum - 1 /* 総数を減じる
6740 continue /* 削除時はiを更新しない
6750 }
6760 i = i + 1
6770 endwhile
6780 return( maxNum ) /* 総数を返す
6790 endfunc
6800 /*
6810 /* cards以外のカードを調べる
6820 /*
6830 func int addCards( maxNum )
6840 int i, num
6850 for i=0 to maxNum-1 /* cards内のカードに対応する
6860 usedCards( cards( i ) ) = 2 /* usedCardsを2にする
6870 next
6880 num = maxNum
6890 for i=0 to 99 /* 0番のカードから順に
6900 if usedCards( i ) = 1 then { /* 使用されているカードで
6910 if findCards( i ) then { /* 条件合致をチェック
6920 cards( num ) = i /* 合致すればcardsに登録
6930 num = num + 1 /* 総数を増やす
6940 }
6950 }
6960 next
6970 for i=0 to maxNum-1 /* 2にしたものを1に戻す
6980 usedCards( cards( i ) ) = 1
6990 next
7000 return( num ) /* 総数を返す
7010 endfunc
7020 /*
7030 /* カードが条件に合致するか調べる
7040 /*
7050 func int findCards( No ) /* Noのカードが合致するか調べる
7060 int flag
7070 int i
7080 str op, cond[97]
7090 fseek( rat, 2100 + No*4*31, 0 ) /* Noまでシークし
7100 fread( dbasep, 31, rat ) /* カードを読み込む
7110 for i=0 to 30
7120 if Data( i ) <> "" then { /* 条件が設定されているなら
7130 flag = 0
7140 op = left$( Data( i ), 1 ) /* 演算子と
7150 cond = mid$( Data( i ), 2, 96 ) /* 比較データを取り出し
7160 fseek( dbase, dbasep(i)*97, 0 )
7170 freads( line$, dbase ) /* カードの該当行を取り出して
7180 switch asc( op ) /* 演算子によって分ける
7190 case '=' /* 等しい
7200 if line$ = cond then flag = 1
7210 break
7220 case '<' /* 小さい
7230 if line$ < cond then flag = 1
7240 break
7250 case '>' /* 大きい
7260 if line$ > cond then flag = 1
7270 break
7280 case '!' /* 等しくない
7290 if line$ <> cond then flag = 1
7300 break
7310 case '{' /* 条件に含まれる
7320 if instr( 1, cond, line$ ) then flag = 1
7330 break
7340 case '}' /* 条件が含まれる
7350 if instr( 1, line$, cond ) then flag = 1
7360 break
7370 endswitch
7380 if condition = 0 then { /* AND検索で
7390 if flag = 0 then { /* 条件不成立なら
7400 break /* 検索終了
7410 }
7420 } else { /* OR検索で
7430 if flag = 1 then { /* 条件成立なら
7440 break /* 検索終了
7450 }
7460 }
7470 }
7480 next
7490 return( flag ) /* 見つかったかどうかを返す
7500 endfunc
7510 /*
7520 /* シェルソート
7530 /*
7540 func shellSort( num, keyLine, sortKey;str )
7550 int gap, i, j, keyln( 0 )
7560 int tmp /* swap用
7570 /*
7580 for i=0 to num /* sortキーの読み込み
7590 faek( rat, 2100 + cards( i ) * 4 * 31 + keyLine * 4, 0 )
7600 fread( keyln, 1, rat )
7610 fseek( dbase, keyln( 0 ) * 97, 0 )
7620 freads( line$, dbase )
7630 keys( i ) = line$
7640 next
7650 /*
7660 gap = num / 2 /* シェルソート開始
7670 locate 0, 31 /* 最下行に
7680 while gap > 0
7690 print gap; "->"; /* gapを表示 (気体め)
7700 for i= gap to num
7710 j = i - gap
7720 while j >= 0
7730 if sortKey = ">" then { /* 並べ変える必要のない場合
7740 if keys( j ) >= keys( j+gap ) then break
7750 } else { /* をここで排除する
7760 if keys( j ) <= keys( j+gap ) then break
7770 }
7780 tmp = cards( j ) /* カードの並べ替え
7790 cards( j ) = cards( j+gap )
7800 cards( j+gap ) = tmp
7810 line$ = keys( j ) /* キーの並べ替え
7820 keys( j ) = keys( j+gap )
7830 keys( j+gap ) = line$
7840 j = j - gap
7850 endwhile
7860 next
7870 gap = gap / 2
7880 endwhile
7890 endfunc

```

C, X-BASICの関数を作成する

Murata Toshiyuki 村田 敏幸

グラフィック関係のサブルーチンを作ってきましたが、これらはC言語やX-BASICのプログラムからも呼び出して使用できるとはいへん便利です。今回はCの関数、X-BASICの外部関数をマシン語で書く場合の方法を解説します。

ここ3回にわたって、グラフィック周りのプリミティブなサブルーチンをいくつか作ってきた。課題扱いだったものも含めてリストで示したものを数え上げると結構な数にのぼる。今月は、これらサブルーチンをC言語、X-BASICから利用する場合を例に、Cの関数をマシン語で書く方法、X-BASICの外部関数の作成方法を見ていこうと思う。

変数の扱い方

予備知識として、Cにおける変数の扱いについてマシン語の立場から眺めてみる。

まず、データ型と記憶領域の大きさの関係を示そう。XCにおける基本データ型と占める領域は、

char	1バイト
short	2バイト
int	4バイト
long	4バイト
float	4バイト
double	8バイト

である。また、ポインタは、それが何を指しているように常に4バイトだ¹⁾。さらに、配列は同じ型のデータが連続したメモリに要素数だけ並んだもの²⁾であり、要素のサイズ×要素数分の領域を占める。

構造体はいくつかのデータ型を寄せ集めてまとめて扱えるようにしたもの³⁾だから、基本的には構造体の各メンバーのサイズを合計しただけの大きさを持つ。ただし、68000では偶数境界をまたいだワード/ロングワードのアクセスが禁止されている関係で、char型のメンバーの直後に1バイトの詰め物が入る場合がある。

さて、Cの変数はその記憶場所がどこにとられるかによって次のように分類される。

1) メモリ上の固定領域にとられるもの(長命: プログラムの実行中ずっと生き続ける)

2) スタック上にとられるもの(短命)

3) レジスタ上にとられるもの(短命)

関数の外部で定義された変数と、関数の内部でとくに静的(static)であると明示された変数は1)、関数内の自動的(auto)な変数は2)、そして、いわゆるレジスタ変数として宣言されたもの(register)は空いているレジスタがあれば3)、そうでなければ2)のパターンとなる。リスト1に示した例では、

a, b, c, d, g, hは 1)
e, x, yは 2)
fは 3)

にそれぞれ該当する。Cでは関数への引数の引き渡しにスタックを利用するので、関数fooの仮引数であるx, yも2)のパターンに含まれることに注目してもらいたい。

マシン語に当てはめて考えると、1)はデータセクションやbss(Block Storage Section: ブロックストレージセクション)⁴⁾に確保したワークエリア、2)はスタック上に確保したワークエリアと、サブルーチン呼び出し時にスタックに積まれた引数、3)はレジスタそのものに相当するわけだ。

通用範囲(scope: スコープ)について分類するなら、次のようになる。

A) プログラム中のどこからでも参照できるもの
イ) 1本のソースファイル中(の定義された行からファイルの終わりまで)でのみ参照できるもの
ウ) 特定のブロック内(“{” ~ “}”)でのみ参照できるもの

関数の外部でstaticをつけずに定義するとA)、関数外部でstaticに定義するとイ)、関数内で定義するとウ)になる。リスト1では、

a, bは ア)
c, dは イ)
e, f, g, h, x, yは ウ)

1) アドレスを保持するのがポインタ変数なのだから。
2) 配列は、dcがずらずら並んだもの、もしくは、.dsで確保された一連のメモリ領域をイメージすればよい。
3) 構造体(のテンプレート)は、offsetによるオフセット表のイメージ。
4) storageは英語の発音に従えば、ストリッジないしはストーリッジと表記されるべきなのだろうが、ここではよく見かける表記を採用した。ちなみに、“(データの)保管場所”とか、もう少し意識すれば“メモリ領域”程度の意味。

5) “_”をつけるのは、コンパイラが自分で勝手に使うラベルとCソース中の識別名がかちあわないようにするため。

6) ここでいうコモンエリアはHuman68k Ver.2.0のconfig.sysのcommon行で指定するものとはまったく別のもの。念のため。

7) リスト2ではたまたま.bssと.endの間に.commが置かれているが、実際には.commはどこに置いてもかまわない。

8) コモンエリア以外のbssは初期化されることはない(初期値はメモリ上のゴミ)。

9) ただし、staticがついていようといなかろうと、初期値の指定がない静的な変数は0で初期化されるはずだから、このコンパイル結果は誤りだと思う。変数cの場合と同様に、.dc.lで領域を確保するのが正しい(.commで定義するのもまた誤り。外部からも参照可能になってしまい、スコープが定義と違ってしまう)。

10) これも本当は0で初期化すべきものだろう。

にそれぞれ該当する。

AS, X仕様のアセンブリ言語に当てはめてみるとア)は.xdef(ないしは.global)で外部宣言されたワーク、イ)ウ)は.xdefで宣言されていない、もしくはスタック上の一時的なワークと考えることができる。また、リスト1には出てきていないが、C言語のextern宣言は、“プログラムのどこかで同名の変数が定義されている”ということをコンパイラに知らせるものだから、ちょうど.xrefに相当する。

実際にリスト1をXC Ver.2.0でコンパイルしてみた結果がリスト2。XCは馬鹿正直なコードを出してくれるので、このような例にはうってつけた。リスト1のa~hの変数定義がどのようにコンパイルされているか見てもらおう。

・変数 a

“外部からも参照可能なメモリ上のintの変数で初期値は1”だから、データセクション中に.dc.lで領域が確保される(26行)と同時に.globalによる外部定義が行われている(2行)。見てのとおり、XCでは、変数に割り当てられたメモリ領域の先頭アドレスを、“Cにおける識別名の頭にアンダーバー(_)をつけたラベル”で表す⁵⁾。

・変数 b

“外部からも参照可能なメモリ上のintの変数で初

期値は指定されていない”というパターンだ。Cでは静的な変数は初期値が指定されなければ0で初期化されることになっており、リスト2では.comm疑似命令によってコモン(common:共通)エリア⁶⁾に4バイト分の0で初期化された領域を確保している(35行)。コモンエリアは、bssの一部⁷⁾で、xファイルがメモリに読み込まれるときにHuman68kによって0で初期化される⁸⁾領域だ。

・変数 c

“このソース内部でのみ通用するメモリ上のintの変数で初期値は1”だ。28行で.dc.lにより領域は確保されているが、外部定義はない。

・変数 d

“このソース内部でのみ通用するメモリ上のintの変数で初期値の指定なし”なので、bssに.ds.bにより4バイトの領域だけ⁹⁾が確保されている(34行)。外部定義もない。

・変数 e

関数内で記憶クラスの指定なしに宣言された変数はautoと見なされ、関数呼び出しのたびにスタックフレーム上に領域が一時的に確保される。リスト2では、7行で4バイト分のローカルエリアが確保され、変数eは“-4(a6)”で参照されているのがわかるだろう。

・変数 f

レジスタ変数だ。XCではレジスタ変数はd3~d7、a3~a5のいずれかに割り当てられる。通常の変数はデータレジスタに、ポインタはアドレスレジスタに、と決まっているらしい。リスト2では変数fはd7レジスタに割り当てられている。

・変数 g

関数内でのみ通用するローカル変数だが、staticがついているから領域はメモリ上の固定領域にとられる。リスト2では37行でbssに4バイトの領域が確保されている¹⁰⁾。ただ、その領域にふられたラベルが“_g”ではなく、“L5”になっている。これは、もし関数の外側でgという別の変数が定義されていても、ラベル定義がぶつからないようにする

リスト1 VARTEST.C

```
1: int a = 1;
2: int b;
3:
4: static void foo( x, y )
5: {
6:     int e;
7:     register int f;
8:     static int g = 1;
9:     static int h;
10:
11:     e = x;
12:     h = y;
13:     f = g;
14: }
15:
16:
17: static int c = 1;
18: static int d;
19:
20: int main()
21: {
22:     foo( c, 50000 );
23:
24:     return 0;
25: }
```

リスト2 VARTEST.S

```
1: INCLUDE fefunc.h
2: .GLOBL _a
3: .GLOBL _main
4: .XREF __main
5: .TEXT
6: _foo:
7:     LINK    A6,#-4
8:     MOVE.L  D7,-(SP)
9:     MOVE.L  8(A6),-4(A6)    *e = x;
10:    MOVE.L  12(A6),L5        *h = y;
11:    MOVE.L  L4,D7            *f = g;
12:    MOVE.L  (SP)+,D7
13:    UNLK    A6
14:    RTS
15: _main:
16:    LINK    A6,#0
17:    MOVE.L  #50000,-(SP)    *
18:    MOVE.L  _c,-(SP)        *
19:    JSR     _foo            *foo( c, 50000 );
```

```
20:    ADDQ.L  #8,SP          *
21:    MOVEQ.L  #0,D0          *return 0;
22:    UNLK    A6              *
23:    RTS                     *
24:
25: _a:
26:    .DC.L   $00000001
27: _c:
28:    .DC.L   $00000001
29: L4:
30:    .DC.L   $00000001
31:    .EVEN
32:    .BSS
33: _d:
34:    .DS.B   4
35:    .COMM   _b,4
36: L5:
37:    .DS.B   4
38:    .END
```


ためだ。変数gはあくまでローカル変数だから、ほかの関数から参照されることはない。

・変数h

ローカルでstaticな変数に初期値を与えた例だ。データセクションに.dc.1で領域が確保されている(30行)。

・変数x, y

関数の仮引数だ。auto変数同様、スタックフレーム上の位置で示されている。リスト2では“8(a6)”が変数x, “12(a6)”が変数yにあたる。

・関数foo, main

ついでに、関数にも目を向けてもらいたい。関数fooはstaticに宣言してある。したがって、.xdefによる外部定義は行われず、ほかのソースから参照することはできない。対して、mainは記憶クラスの指定がないので、3行で外部定義が行われている。

余談ながら、4行で外部参照定義された__mainというラベルは、Cのスタートアップルーチンの先頭アドレス、つまりはこのCプログラムの実行開始アドレスを意味している¹¹⁾。この定義があることで、リンク時にリンカは__mainというシンボルをライブラリから探し、結果としてスタートアップルーチンをリンクすることになるわけだ。

C言語の関数をマシン語で

そろそろCの関数をアセンブリ言語で書く方法を考えてみよう。上で示した情報から、いくつかの約束ごとが見えてくる。

- 1) 関数は、Cで使う関数名の頭に“_”をつけたラベル名で表されるサブルーチンとして記述する。
- 2) そのサブルーチン名は.xdefまたは.globlで外部定義しておく。
- 3) 引数の受け渡しはスタックを介して行う。
- 4) 呼び出し元の関数でレジスタ変数を使っているかもしれないから、サブルーチン内でd3~d7, a3~a5を使うときには値を保存しておく。

```
1:      .include      iocscall.mac
2:      *
3:      .xdef      _gcopy
4:      .xref      gcopy
5:      *
6:      .offset      8
7:      *
8:      X0:      .ds.l      1
9:      Y0:      .ds.l      1
10:     X1:      .ds.l      1
11:     Y1:      .ds.l      1
12:     X2:      .ds.l      1
13:     Y2:      .ds.l      1
14:     *
15:     .text
16:     .even
17:     *
18:     _gcopy:
19:     link      a6,#0
20:
21:     suba.l    a1,a1
22:     IOCS      _B_SUPER
```

5) 当然、フレームポインタであるa6とスタックポインタであるa7 (sp) の値も保存する必要がある。

また、リスト1と2を注意深く見比べると、次のようなこともわかる。

6) 関数へ渡す引数をスタックに積む順序は、Cにおける引数の並び順の逆(リスト2の17~18行)

7) 関数の戻り値はd0レジスタに入れて返す(リスト2の21行)¹²⁾。ただし、fooのようなvoid型の(値を返さない)関数では戻り値の心配をする必要はない。

では、試しに先月のサブルーチンgcopyをCから呼び出せるようにしてみよう。単に呼び出せるようにするだけであれば、非常に簡単だ。先月のリスト10 (GCOPY.S) 中、4行の、

```
.xdef      gcopy
```

の直後に、

```
.xdef      _gcopy
```

を、また、22行の、

```
gcopy :
```

の直後に、

```
_gcopy :
```

を挿入し、アセンブルし直せば、サブルーチン_gcopy (実体はサブルーチンgcopyと同じもの) がgcopy() という名前の関数として呼び出せる形式になる。

ただ、サブルーチンgcopyは転送元/転送先の領域の座標を任意のメモリ領域に書き込み、その先頭アドレスをスタックに積んで呼び出すようにできており、これをCで実現するにはリスト3のように構造体に引数を代入して、その構造体の先頭アドレスを関数に渡ししかない。また、gcopy内ではスーパーバイザ空間にアクセスすることになるので、gcopy() を呼び出す前にあらかじめスーパーバイザモードへ移行しておく必要もある。仕様です、と突っばねることもできるが、どうせCの関数にするのなら、

11) Cでは一部の関数の初期化のために、関数mainの実行に先立って、このスタートアップルーチンが実行される。関数mainはその中からサブルーチンコールされることになる。

12) doubleや構造体を戻り値とする関数ではまた違ってくるのだが、ここでは触れない。

リスト3 SMPL.C

```
1: int main()
2: {
3:     struct GCOPYBUF {
4:         short x0, y0;
5:         short x1, y1;
6:         short x2, y2;
7:     } gcopybuf;
8:
9:
10:
11:     gcopybuf.x0 = 0;
12:     gcopybuf.y0 = 0;
13:     gcopybuf.x1 = 50;
14:     gcopybuf.y1 = 50;
15:     gcopybuf.x2 = 100;
16:     gcopybuf.y2 = 100;
17:     gcopy( &gcopybuf );
18:
19:
20: }
```

リスト4 _GCOPY.S

```
23:      move.w    Y2+2(a6),-(sp)
24:      move.w    X2+2(a6),-(sp)
25:      move.w    Y1+2(a6),-(sp)
26:      move.w    X1+2(a6),-(sp)
27:      move.w    Y0+2(a6),-(sp)
28:      move.w    X0+2(a6),-(sp)
29:
30:
31:      move.l    sp,-(sp)
32:      jsr      gcopy
33:
34:      tst.l     d0
35:      bmi      done
36:
37:      movea.l   d0,a1
38:      IOCS      _B_SUPER
39:
40: done:      unlk      a6
41:      rts
42:
43:      .end
```


gcopy(0, 0, 50, 50, 100, 0);
のような素直な形式で、ユーザーモードからも呼び出せるようにしたいところだ。

そこで、また別の方法を検討する。サブルーチンgcopyとCプログラムとの間にワンクッション入れて、つじつまを合わせることにしよう。リスト4だ。サブルーチンの頭でスーパーバイザモードへ移行し、スタック上に積まれた座標値をサブルーチンgcopyに渡せる形に再構成してからgcopyをコールし、そののち再びB_SUPERでユーザーモードに戻している。

スーパーバイザモードへの移行にはいつものDOSコールsuperではなく、IOCSコールのB_SUPERを使ってみた。特に機能的な違いはない。ただ、superではスタックを介して引数の受け渡しを行うのに対し、B_SUPERではスタックの代わりにa1レジスタが使われる。なお、B_SUPERはすでにスーパーバイザモードなのにさらにスーパーバイザモードにしようとした場合にはd0に-1を返すことになっているので、34~35行でそのチェックをしている。関数が呼び出された時点ですでにスーパーバイザモードであれば、そのまま呼び出し元に戻るわけだ。

引数の再構成に関しては、特に説明するまでもないだろう。スタック上にロングワード単位で積まれ

た6つの値を、ワード単位でもう一度スタックに積み直している。その時点で、スタック上にサブルーチンgcopyの引数受け渡し領域と同じ構造のデータが出来上がるから、その先頭アドレス(=sp)をgcopyに引き渡せばよい。

と、こうして作成した関数gcopy()を使うときの注意点を挙げておこう。まず、当然、必要なモジュールをすべてリンクしてやらなければならない。_GCOPY.O(リスト4をアセンブルしたもの)とGCOPY.O(先月のリスト10をアセンブルしたもの)、さらにはGCOPY.S内で外部参照されているいくつかのサブルーチン、データも忘れずにリンクする。実用上は、これら(+なんならここ4回分の全サブルーチン)をひとまとめのライブラリしておくのが便利だろう。

また、gcopy()内ではグラフィック画面が初期化されているかどうか、画面モードがどうなっているかのチェックは一切行っていない。画面の初期化は呼び出し元のCプログラム側で管理すること。

さらに、描画ページの切り替えの問題もある。gcopyは描画ページを自前で管理しており、XCのapage()の影響を受けない。この問題は、リスト5のような関数(サブルーチン)を元のapage()の代わりに使うことで回避できる。

ああ、まだあった。gcopyはアセンブル時に実画面1024×1024ドットモード用か、512×512ドットモード用かを決定するように作られていたのだった。1024ドットモード用と、512ドットモード用の2つのライブラリを作っておき、プログラムで使う画面モードに応じてリンクするライブラリを変えるなどの方法で逃げてもらいたい。

“Cの関数をマシン語で書く方法”だか、“最近作ったサブルーチンをCから使えるようにつじつま合わせする方法”だかわからなくなったところで、この話題は終わりとする。gcopy以外のほかのサブルーチンもまったく同じ方法でCの関数にできるから、

リスト5 _APAGES

```
1:      .include      iocscall.mac
2:      *
3:      .xdef      _apage
4:      .xref      apage
5:      *
6:      .text
7:      .even
8:      *
9:      _apage:
10:     arg =      4+3
11:     move.b      arg(sp),d1
12:     bmi      getapage
13:     ext.w      d1
14:     jmp      apage      *以前作ったサブルーチン
15:                                     *引数はd1.wにページ番号
16:     *
17:     getapage:
18:     IOCS      _APAGE
19:     rts
20:
21:     .end
```

リスト6 GRAPH2.H

```
1: /* graph2.h */
2:
3: #if !defined( __GRAPH2 )
4: #define __GRAPH2
5:
6: #define _INT4      int, int, int, int
7: #define _INT5      int, int, int, int, int
8: #define _INT6      int, int, int, int, int, int
9: #define _INT4A      int, int, int, void *
10: #define _INT4AI      int, int, int, int, void *, int
11:
12: void gfill( _INT5 );
13: void gfill_or( _INT5 );
14: void gfill_and( _INT5 );
15: void gfill_xor( _INT5 );
16: void gtilefill( _INT6 );
17: void gnegate( _INT4 );
18: void gmonotone( _INT4 );
19: void gmonotone_y( _INT4 );
20: void gsoftfocus( _INT4 );
21: void goutline( _INT4 );
22: void ghalftone( _INT4 );
```

```
23: void gmonotone_hsv( _INT6 );
24: void gmonotone_hsy( _INT6 );
25: void gaddcolor( _INT5 );
26: void gsubcolor( _INT5 );
27: void gget( _INT4A );
28: void gput( _INT4A );
29: void gputon( _INT4AI );
30: void gputin( _INT4AI );
31: void ghalftoneput( _INT4A );
32: void gcopy( _INT6 );
33: void ghreverse( _INT4 );
34: void gvreverse( _INT4 );
35: void ghdup( _INT5 );
36: void gvdup( _INT5 );
37:
38: #undef _INT4
39: #undef _INT5
40: #undef _INT6
41: #undef _INT4A
42: #undef _INT4AI
43:
44: #endif
```


各自やってみてもらいたい。おまけとして、リスト6に“そうやって作られるであろう”グラフィックライブラリ用のプロトタイプ宣言を含むヘッダファイルを示しておく。

で、次はX-BASICだ。

X-BASICの外部関数

X-BASICの外部関数の作成方法については『プログラマーズマニュアル』に詳しいし、本誌にもときおり思い出したように解説記事が載る。あえて補足することはなにもないのだが、とりあえず実例をひとつ示しておきたい。

半ばやけくそ気味で、最近作ったグラフィック関連サブルーチンを全部まとめたGRAPH2.FNCを作ってみる。ただし、この外部関数は画面モードが512×512ドット65536色モードでのみ使用可能とする。

X-BASICの外部関数ファイルはXファイルと同じ形式をとる。作成手順もふつうのマシン語プログラムと変わらない。ただし、プログラムの頭の部分にインフォメーションテーブルと呼ばれるヘッダをつけることになっている(図1)。実例を見ないとピンとこないかもしれないが、簡単に解説しておこう。

インフォメーションテーブルの頭の部分にはなにか特別なイベントが発生したときに呼び出される6つのルーチンの先頭アドレスが並ぶ。これらのすべてをサポートする必要はないが、サポートしないものについてはrts命令が置かれたアドレスを格納しておかなければならない¹³⁾。続く2ロングワードは将来の拡張用で、いまのところはrtsが置かれたアドレスを入れておけばよい。

その後ろに、トークンテーブル、パラメータテーブル、関数の実行アドレステーブルの各先頭アドレスが続く。トークンテーブルは外部関数ファイル内の全関数名を0で終わる文字列の形で並べたもの。テーブルの最後を示す意味でトークンテーブルの末尾にはもう1個余分な0を置く。

パラメータテーブルにはポインタが並び、“そのポインタの指す先”に、パラメータIDとか呼ばれているコードで各関数の引数/戻り値の型情報を置く。パラメータIDはひとつの引数/戻り値あたり16ビットで、各ビットには一応図2に示すような意味が割り振られている¹⁴⁾。なお、XCにはFDEF.Hというファイルがついてきて、よく使われるパラメータIDがシンボル定義されている。

実行アドレステーブルは言葉どおり各関数の処理ルーチン本体の先頭アドレスを並べたものだ。

ここで、トークンテーブル、パラメータテーブル、実行アドレステーブル内の関数の並びはすべて同じ順序でなければならない。

さて、X-BASICの外部関数への引数受け渡しにはやはりスタックが用いられ、C同様、引数の並びの後ろから順に積まれる。ただし、スタックには引数の値に加えて、各引数の型、引数の総個数が積まれる(図3)。ひとつの引数あたり10バイト、うち、値の格納用に8バイトが割り当てられているわけだが、実際に8バイトすべてが使われるのはfloat型の場合だけで、int型の場合は下位4バイトのみ、char型の場合は下位の1バイトのみが使われ、残り

13) 外部関数とそのイベントをサポートしているかどうかに関わらず、X-BASICインタプリタはイベント発生時に無条件にこのテーブルを参照し、該当アドレスをサブルーチンコールする。

14) ただし、X-BASICはすべての組み合わせをサポートしているわけではないようだ。

図1 X-BASIC外部関数インフォメーションテーブル

+00 _H	1L	X-BASIC起動時に呼び出されるサブルーチン先頭アドレス
+04 _H	1L	RUN命令実行時に呼び出されるサブルーチン先頭アドレス
+08 _H	1L	END命令実行時に呼び出されるサブルーチン先頭アドレス
+0C _H	1L	SYSTEM命令、EXIT()関数実行時に呼び出されるサブルーチン先頭アドレス
+10 _H	1L	BREAK、^Cによりプログラムが中断されたときに呼び出されるサブルーチン先頭アドレス
+14 _H	1L	^D入力時に呼び出されるサブルーチン先頭アドレス
+18 _H	1L	予備
+1C _H	1L	予備
+20 _H	1L	トークンテーブル先頭アドレス
+24 _H	1L	パラメータテーブル先頭アドレス
+28 _H	1L	実行アドレステーブル先頭アドレス
+2C _H	5L	予備(0で埋めておく)
...		
+3F _H		

図2 X-BASIC外部関数パラメータID

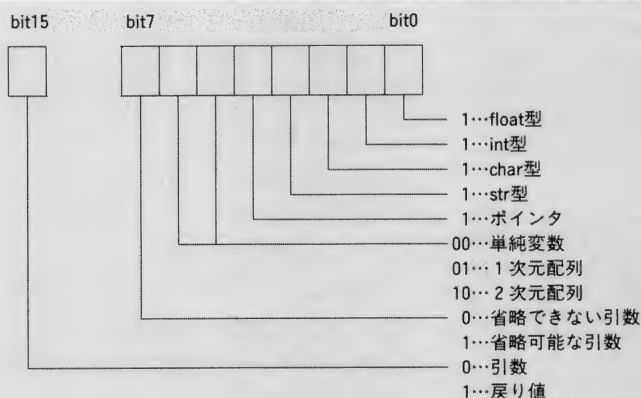


図3 X-BASIC外部関数へ渡される引数

SP+00 _H	1L	リターンアドレス
SP+04 _H	1W	引数の総個数
SP+06 _H	1W	第1引数の型
SP+08 _H	1L	第1引数の値(上位4バイト)
SP+0C _H	1L	第1引数の値(下位4バイト)
SP+10 _H	1W	第2引数の型
SP+12 _H	1L	第2引数の値(上位4バイト)
SP+16 _H	1L	第2引数の値(下位4バイト)
SP+1A _H	1W	第3引数の型

図4 配列ポインタの構造(1次元配列の場合)

+00 _H	1L	X-BASIC内部で使用
+04 _H	1W	配列の次元数-1
+06 _H	1W	配列要素1個のバイト数
+08 _H	1W	配列の要素数-1(添え字の最大値)
+0A _H		配列の内容

は0で埋められる。また、str型/配列の場合は下位4バイトにデータへのポインタが格納されている。このポインタはstr型の場合は文字列の先頭を指すが、配列の場合は図4に示すような構造を持ったメモリ領域を指す。なお、図4は1次元配列の場合で、多次元配列の場合はもう少し情報量が増える。

関数からの戻り値は引数と同じような構造の10バイトの領域（ただし、型の部分は0にする）に値を収め、その先頭アドレスをa0に入れて返す。加えて、エラーなく処理が終了した場合にはd0.1に0を、エラーの場合は0以外のエラーコードを、a1にエラーメッセージへのポインタを返すことになっている。ただし、負のエラーコードにはHuman68kのDOSコールと同等の意味が割り当てられているようで、特にa1で指定しなくても適当なメッセージが表示されるようだ。外部関数特有のエラーメッセージを表示したい場合は、正のエラーコードを使えばよい。なお、正常終了時はa1は無視され、エラー時にはa0が¹⁵⁾無視される。

では、リスト7を見てもらおう。各関数の実行ルーチン本体は含まれていない。引数や戻り値のつじつま合わせだけを行い、処理自体はすでに作ってあるサブルーチンにまかせるようになっている。

まず、インクルードファイルを取り込んでいる。FDEF.HはXCのものを流用しているが、もし不幸にしてFDEF.Hが手元にない人のために7～9行に注釈の形で必要部分を抜粋してある。続いて、外部参照がずらっと並ぶ。この外部関数を作成するには、ここに並んだすべてのサブルーチンをリンクする必要があるわけだ。たぶん、読者の多くはすべてのサブルーチンを入力しているほど元気じゃないと思うから、逃げ道を示しておこう。

- 1) 用意していないサブルーチンの外部参照を削る（たとえば、gfillがなければ13行を削除する）
- 2) そのサブルーチン名を387行の直前に挿入して、rtsだけのサブルーチンにしてしまう（gfillの例だと、

gfill:

の1行を追加することになる)

スタックフレームなどのデータ構造のオフセット表定義に続いて、76行からがインフォメーションテーブルだ。イベントに対応した処理ルーチンは使う必要がなかったのだ、どれもrtsを指している。90行からのトークンテーブルは例の偶数境界の問題を避ける意味で最後にeven疑似命令が置いてある¹⁶⁾。

その後ろ、121行以降がパラメータテーブル。先に触れたとおり、パラメータテーブルには“関数の引数/戻り値の個数/型を表す情報へのポインタ”を並べる。この外部関数ファイル中では引数の数や型

が同一の関数が多いので、パラメータID列もある程度共通化してメモリを節約している。ちなみに、どの関数も戻り値はintで、正常終了時は0、エラー時は1を関数の戻り値とする（と決めた）¹⁷⁾。

176行からの実行アドレステーブルはいいとして、207行から始まる各関数の処理ルーチンを見てもらおう。関数の多くは“引数の数”と“どの既存サブルーチン呼び出すか”が異なるだけなので、処理の多くの部分は（紙面の無駄遣いを避ける意味で）やはり共通化してあり、なんだかんだでほとんどの関数は367行で合流している（その分braだらけになってしまったが）。

368～377行がエラーチェックを行っている部分で、IOCSコールAPAGEでグラフィック画面が初期化されているかどうかと、IOCSコールCRTMODにより画面モードが65536色モードかどうかを確認し、違っていたらその場でエラー処理ルーチンに飛んでエラー終了する。この時点でスタックにはいくつもの値が積みっ放しになっているが、396行のunlkで破算になるから大丈夫。

ちょっと戻って、重要どころの配列の扱いを見てもらおう。329行以下の部分だ。gget()やgput()などの関数はパターン¹⁸⁾の格納領域として、任意の数値型の1次元配列を受け取る。このとき、その配列は、指定されたG-RAM上の領域を十分格納できるだけの大きさを持っていないといけない。配列の大きさは引数情報中の要素ひとつあたりのバイト数×要素数で計算できる。また、G-RAM上の領域が何バイト分かは縦横のドット数の積×2で得られる。比較の結果配列のバイト数のほうが小さければエラーだ¹⁸⁾。

あと、パレットをまとめてセーブ/ロードする関数gsavepalet(), gloadpalet()は引数としてファイル名をとる形にした(292～314行)。元のサブルーチンはファイルハンドルを引数とするようになっていたが、X-BASICのファイル番号はファイルハンドルとは似てはいるが異なるものなので、外部関数の中でファイルのオープンからクローズまでをまとめてやってしまっている。期せずして、str型の引数をとる関数の単純な例になった。

以上、今回は高級言語とマシン語をリンクする方法を見てきた。この連載でこんなことをいってはずいぶん気分が落ちるが、速度が要求される部分だけをマシン語で書いてメイン部分は高級言語で書くという手は、開発効率の点ではなかなかおいしい。逆に、マシン語の知識を持つことで高級言語の使い道にも幅が出てくる、といういい方もできるかもしれない。と強引に締めたところで、また来月。

15) error offによりエラーを無視する設定になっているときには、エラー時にもa0は意味を持つ。

16) 実際に文字列が何文字かは数えていない。もし、もともとトークンテーブルが偶数バイトだった場合にはevenは単に無視されるだけだからだ。

17) ここでいっている関数の戻り値はd0に返すエラーコードではなく、X-BASICプログラム側から見た関数の戻り値だが、結局はd0にも同じ値を返すように作ってある。

18) gget()はちょっと変な仕様にしてしまった関係で本来なら別扱いにするべきなのだが、手を抜いてgput()などと処理を共有している。


```

1:      .include      doscall.mac
2:      .include      iocscall.mac
3:      .include      fdef.h
4:      .include      const.h
5:      .include      gmacro.h
6:      *
7:      *int_val      equ      $0002      *int型の引数
8:      *str_val      equ      $0008      *str型の引数
9:      *aryl_fic      equ      $0037      *数値型1次元配列
10:     *int_ret      equ      $8001      *int型の戻り値
11:     IOCS_GL3      equ      12
12:     *
13:     .xref      gfill
14:     .xref      gfill_or
15:     .xref      gfill_and
16:     .xref      gfill_xor
17:     .xref      gtilefill
18:
19:     .xref      gsavapalet
20:     .xref      gloadpalet
21:     .xref      gnegate
22:     .xref      gmonotone
23:     .xref      gmonotone_y
24:     .xref      gsoftfocus
25:     .xref      goutline
26:     .xref      ghalftone
27:     .xref      gmonotone_hsv
28:     .xref      gmonotone_hsy      *
29:     .xref      gaddcolor
30:     .xref      gsubcolor
31:
32:     .xref      gget
33:     .xref      gput
34:     .xref      gputon
35:     .xref      gputin
36:     .xref      ghalftoneput
37:     .xref      gcopy
38:     .xref      ghreverse
39:     .xref      gvreverse
40:     .xref      ghdup
41:     .xref      gvdup
42:     *
43:     .offset 0      *スタックフレーム
44:     *
45:     A6BUF:      .ds.l      1
46:     RETADR:      .ds.l      1
47:     PARC:      .ds.w      1
48:     PAR1:      .ds.b      10
49:     PAR2:      .ds.b      10
50:     PAR3:      .ds.b      10
51:     PAR4:      .ds.b      10
52:     PAR5:      .ds.b      10
53:     PAR6:      .ds.b      10
54:     PAR7:      .ds.b      10
55:     *
56:     .offset 0      *パラメータバッファ
57:     *
58:     TYPE:      .ds.w      1      *型
59:     FVAL:      .ds.l      1      *実数
60:     PVAL:
61:     LVAL:      .ds.w      1      *32ビット数, ポインタ
62:     WVAL:      .ds.b      1      *16ビット数
63:     BVAL:      .ds.b      1      *8ビット数
64:     *
65:     .offset 0      *X-BASICの配列
66:     *
67:     ASKIP:      .ds.w      1
68:     ADIM:      .ds.w      1      *次元数-1
69:     ASIZ:      .ds.w      1      *1要素のバイト数
70:     ALEN:      .ds.w      1      *要素数-1
71:     ADAT:      *データ本体
72:     *
73:     .text
74:     .even
75:     *
76:     information_table:
77:     .dc.l      dummy      *X-BASIC起動時
78:     .dc.l      dummy      *run
79:     .dc.l      dummy      *end
80:     .dc.l      dummy      *system,exit
81:     .dc.l      dummy      *BREAK,^C
82:     .dc.l      dummy      *^D
83:     .dc.l      dummy      *予備
84:     .dc.l      dummy      *予備
85:     .dc.l      token_table
86:     .dc.l      param_table
87:     .dc.l      exec_table
88:     .dc.l      0,0,0,0      *予備
89:     *
90:     token_table:
91:     .dc.b      'gfill',0
92:     .dc.b      'gfill_or',0
93:     .dc.b      'gfill_and',0
94:     .dc.b      'gfill_xor',0
95:     .dc.b      'gtilefill',0
96:     .dc.b      'gsavapalet',0
97:     .dc.b      'gloadpalet',0
98:     .dc.b      'gnegate',0
99:     .dc.b      'gmonotone',0
100:    .dc.b      'gmonotone_y',0
101:    .dc.b      'gsoftfocus',0
102:    .dc.b      'goutline',0

```

```

103:    .dc.b      'ghalftone',0
104:    .dc.b      'gmonotone_hsv',0
105:    .dc.b      'gmonotone_hsy',0
106:    .dc.b      'gaddcolor',0
107:    .dc.b      'gsubcolor',0
108:    .dc.b      'gget',0
109:    .dc.b      'gput',0
110:    .dc.b      'gputon',0
111:    .dc.b      'gputin',0
112:    .dc.b      'ghalftoneput',0
113:    .dc.b      'gcopy',0
114:    .dc.b      'ghreverse',0
115:    .dc.b      'gvreverse',0
116:    .dc.b      'ghdup',0
117:    .dc.b      'gvdup',0
118:    .dc.b      0      *テーブル終端
119:    .even
120:    *
121:    param_table:
122:    .dc.l      param_5i      *gfill
123:    .dc.l      param_5i      *gfill_or
124:    .dc.l      param_5i      *gfill_and
125:    .dc.l      param_5i      *gfill_xor
126:    .dc.l      param_6i      *gtilefill
127:    .dc.l      param_str      *gsavapalet
128:    .dc.l      param_str      *gloadpalet
129:    .dc.l      param_4i      *gnegate
130:    .dc.l      param_4i      *gmonotone
131:    .dc.l      param_4i      *gmonotone_y
132:    .dc.l      param_4i      *gsoftfocus
133:    .dc.l      param_4i      *goutline
134:    .dc.l      param_4i      *ghalftone
135:    .dc.l      param_6i      *gmonotone_hsv
136:    .dc.l      param_6i      *gmonotone_hsy
137:    .dc.l      param_5i      *gaddcolor
138:    .dc.l      param_5i      *gsubcolor
139:    .dc.l      param_4ia      *gget
140:    .dc.l      param_4ia      *gput
141:    .dc.l      param_4iai      *gputon
142:    .dc.l      param_4iai      *gputin
143:    .dc.l      param_4ia      *ghalftoneput
144:    .dc.l      param_6i      *gcopy
145:    .dc.l      param_4i      *ghreverse
146:    .dc.l      param_4i      *gvreverse
147:    .dc.l      param_5i      *ghdup
148:    .dc.l      param_5i      *gvdup
149:    *
150:    param_6i:      .dc.w      int_val      *引数はint 6個
151:    param_5i:      .dc.w      int_val      *引数はint 5個
152:    param_4i:      .dc.w      int_val      *引数はint 4個
153:    .dc.w      int_val
154:    .dc.w      int_val
155:    .dc.w      int_val
156:    .dc.w      int_ret      *戻り値はint
157:
158:    param_4ia:      .dc.w      int_val      *引数はint 4個 + 配列
159:    .dc.w      int_val
160:    .dc.w      int_val
161:    .dc.w      int_val
162:    .dc.w      aryl_fic
163:    .dc.w      int_ret      *戻り値はint
164:
165:    param_4iai:      .dc.w      int_val      *引数はint 4個 + 配列 + int
166:    .dc.w      int_val
167:    .dc.w      int_val
168:    .dc.w      int_val
169:    .dc.w      aryl_fic
170:    .dc.w      int_val
171:    .dc.w      int_ret      *戻り値はint
172:
173:    param_str:      .dc.w      str_val      *引数はstr 1個
174:    .dc.w      int_ret      *戻り値はint
175:    *
176:    exec_table:
177:    .dc.l      x_gfill
178:    .dc.l      x_gfill_or
179:    .dc.l      x_gfill_and
180:    .dc.l      x_gfill_xor
181:    .dc.l      x_gtilefill
182:    .dc.l      x_gsavapalet
183:    .dc.l      x_gloadpalet
184:    .dc.l      x_gnegate
185:    .dc.l      x_gmonotone
186:    .dc.l      x_gmonotone_y
187:    .dc.l      x_gsoftfocus
188:    .dc.l      x_goutline
189:    .dc.l      x_ghalftone
190:    .dc.l      x_gmonotone_hsv
191:    .dc.l      x_gmonotone_hsy
192:    .dc.l      x_gaddcolor
193:    .dc.l      x_gsubcolor
194:    .dc.l      x_gget
195:    .dc.l      x_gput
196:    .dc.l      x_gputon
197:    .dc.l      x_gputin
198:    .dc.l      x_ghalftoneput
199:    .dc.l      x_gcopy
200:    .dc.l      x_ghreverse
201:    .dc.l      x_gvreverse
202:    .dc.l      x_ghdup
203:    .dc.l      x_gvdup
204:    *

```



```

205: *      引数がint 4 個の関数
206: *
207: x_gnegate:    lea.l    gnegate,a5
208:                bra     exec_4i
209: *
210: x_gmonotone:  lea.l    gmonotone,a5
211:                bra     exec_4i
212: *
213: x_gmonotone_y: lea.l    gmonotone_y,a5
214:                bra     exec_4i
215: *
216: x_gsoftfocus: lea.l    gsoftfocus,a5
217:                bra     exec_4i
218: *
219: x_goutline:   lea.l    goutline,a5
220:                bra     exec_4i
221: *
222: x_ghalfitone: lea.l    ghalfitone,a5
223:                bra     exec_4i
224: *
225: x_ghreverse:  lea.l    ghreverse,a5
226:                bra     exec_4i
227: *
228: x_gvreverse:  lea.l    gvreverse,a5
229:                bra     exec_4i
230: *
231: *      引数がint 5 個の関数
232: *
233: x_gfill:      lea.l    gfill,a5
234:                bra     exec_5i
235: *
236: x_gfill_and:  lea.l    gfill_and,a5
237:                bra     exec_5i
238: *
239: x_gfill_or:   lea.l    gfill_or,a5
240:                bra     exec_5i
241: *
242: x_gfill_xor:  lea.l    gfill_xor,a5
243:                bra     exec_5i
244: *
245: x_gaddcolor:  lea.l    gaddcolor,a5
246:                bra     exec_5i
247: *
248: x_gsubcolor:  lea.l    gsubcolor,a5
249:                bra     exec_5i
250: *
251: x_ghdup:      lea.l    ghdup,a5
252:                bra     exec_5i
253: *
254: x_gvdup:      lea.l    gvdup,a5
255:                bra     exec_5i
256: *
257: *      引数がint 6 個の関数
258: *
259: x_gtilefill:  lea.l    gtilefill,a5
260:                bra     exec_6i
261: *
262: x_gmonotone_hsv: lea.l    gmonotone_hsv,a5
263:                bra     exec_6i
264: *
265: x_gmonotone_hsy: lea.l    gmonotone_hsy,a5
266:                bra     exec_6i
267: *
268: x_gcopy:      lea.l    gcopy,a5
269:                bra     exec_6i
270: *
271: *      引数がint 4 個 + 1 次元配列の関数
272: *
273: x_gget:       lea.l    gget,a5
274:                bra     exec_4ia
275: *
276: x_gput:       lea.l    gput,a5
277:                bra     exec_4ia
278: *
279: x_ghalfitoneput: lea.l    ghalfitoneput,a5
280:                bra     exec_4ia
281: *
282: *      引数がint 4 個 + 1 次元配列の関数 + int 1 個の関数
283: *
284: x_gputon:     lea.l    gputon,a5
285:                bra     exec_4iai
286: *
287: x_gputin:     lea.l    gputin,a5
288:                bra     exec_4iai
289: *
290: *      バレットのロード/セーブ
291: *
292: x_gsavepalet: link     a6,#0
293:                move.w  #ARCHIVE,-(sp)
294:                move.l  PAR1+LVAL(a6),-(sp)
295:                DOS     _CREATE
296:                move.w  d0,-(sp)
297:                bmi     done
298:                jsr     gsavepalet
299:                DOS     _CLOSE
300:                bra     okret
301: *
302: x_gloadpalet: move.w  #ROPEN,-(sp)
303:                move.l  PAR1+LVAL(a6),-(sp)
304:                DOS     _OPEN
305:

```

```

310:                move.w  d0,-(sp)
311:                bmi     done
312:                jsr     gloadpalet
313:                DOS     _CLOSE
314:                bra     okret
315: *
316: exec_4iai:    link     a6,#0
317:                move.w  PAR6+WVAL(a6),-(sp)
318:                bra     exec4a
319: *
320: exec_4ia:     link     a6,#0
321: exec4a:       movea.l  PAR5+LVAL(a6),a0
322:                move.w  PAR4+WVAL(a6),d3
323:                move.w  PAR3+WVAL(a6),d2
324:                move.w  PAR2+WVAL(a6),d1
325:                move.w  PAR1+WVAL(a6),d0
326:                pea.l   ADAT(a0)
327:                movem.w d0-d3,-(sp)
328:                MINMAX  d2,d0      *x0<x1を保证
329:                MINMAX  d3,d1      *y0<y1を保证
330:                addq.w  #1,d1      *
331:                addq.w  #1,d3      *
332:                sub.w   d0,d2      *d2.w = x1-x0+1
333:                sub.w   d1,d3      *d3.w = y1-y0+1
334:                mulu.w  d3,d2      *d2.l = 領域のドット数
335:                add.l   d2,d2      *d2.l = 領域のバイト数
336:                move.w  ALEN(a0),d0
337:                addq.w  #1,d0      *d0.w = 配列の要素数-1
338:                mulu.w  ASIZ(a0),d0
339:                *          *d0.l = 配列のバイト数
340:                cmp.l   d2,d0      *配列のほうが小さければ
341:                bcs     error      * エラー
342:                bra     exec
343: *
344: exec_6i:      link     a6,#0
345:                move.w  PAR6+WVAL(a6),-(sp)
346:                bra     exec5
347: *
348: exec_5i:      link     a6,#0
349:                move.w  PAR5+WVAL(a6),-(sp)
350:                bra     exec4
351: *
352: exec_4i:      link     a6,#0
353:                move.w  PAR4+WVAL(a6),-(sp)
354:                move.w  PAR3+WVAL(a6),-(sp)
355:                move.w  PAR2+WVAL(a6),-(sp)
356:                move.w  PAR1+WVAL(a6),-(sp)
357:                moveq.l #-1,d1      *グラフィック画面は
358:                IOCS    _APAGE      *
359:                tst.b   d0          * 初期化されているか?
360:                bmi     error
361:                moveq.l #-1,d1      *画面モードは
362:                IOCS    _CRTMOD      *(d0=12,13...65536色モード)
363:                andi.b  #$fe,d0      *(第0ビットクリア)
364:                cmpi.b  #IOCS_GL3,d0
365:                bne     error      * 65536色モードか?
366:                suba.l  a1,a1
367:                IOCS    _B_SUPER
368:                move.l  sp,-(sp)
369:                jsr     (a5)
370:                tst.l   d0
371:                bmi     done
372:                movea.l d0,a1
373:                IOCS    _B_SUPER
374:                okret: moveq.l #0,d0
375:                done:  lea.l  retval,a0
376:                move.l  d0,LVAL(a0)
377:                unlk    a6
378:                dummy: rts
379:                error: lea.l  errmes(pc),a1
380:                moveq.l #1,d0
381:                bra     done
382:                errmes: .dc.b  'in GRAPH2.FNC',0
383:                .even
384:                .data
385:                .even
386:                retval: .dc.w  0
387:                .dc.l  0,0
388:                *
389:                .end

```


C Compiler

C compiler PRO-08K (XC) は、X68000のプログラミング環境を支える総合開発セットです。特に重要なのはX68000のハードウェアを隅々までサポートした豊富なライブラリ群で、話題のGNU Cコンパイラを使用する場合にもこれらのライブラリがなくてはなりません。また、X-BASICのプログラムをコンパイルできるのも魅力です。そのためのライブラリも用意され、これは逆にC言語からBASICの関数が利用できるといったメリットもあるわけです。そしてXCはバージョン2となり、プログラミングを支える環境がさらに強化されました。その目玉がプログラムを実行しながらデバッグができるソースコードデバッガSCD.Xであり、また、モジュール化された大規模プログラム開発の際に役立つファイル保守ユーティリティMAKE.Xなどです。今回はC言語そのものの魅力とともに、XCによってもたらされるプログラミングの世界を初心者の方の皆さんにもわかりやすく紹介していきましょう。

CONTENTS

Cコンパイラのアウトライン	
XC ver.2.0ガイドマップ	荻窪 圭 106
XCを支援するおいしいツール	
ソースコードデバッガを使ってみよう	泉 大介 112
貴方のプログラミングを支援する	
縁の下のプリプロセッサ	中森 章 116
基礎知識からプログラミングへ	
Cライブラリ活用の手引き	丹 明彦 121
多数のソースファイルを管理する	
XCにMAKEが付いてきた	中森 章 127
MIDI制御が加わった	
新しい音楽ドライバOPMDRV2.X	西川善司 129
BASICコンパイラ派に贈る便利ソフト	
XBAStoC CHECKER	西川善司 131
コラム Cコンパイラが扱うファイル	111
使って便利な外部コマンド	115

Cコンパイラのアウトライン

XC ver.2.0ガイドマップ

Ogikubo Kei 荻窪 圭

スーパーヘビー級の開発セットC compiler PRO-68K ver.2.0。一口にXCといっても1本のツールではない。2枚組のシステムディスクの中には、それはもうたくさんのファイルが詰まっているのだ。まずはこれらの中身を探索してみよう。

あまりの重さに耐えかねたのか、組み上げると取っ手の出来上がるかさばる箱になったXC ver.2.0。最近あのタイプの箱は流行っているのか、DynaBook用パーソナルプリンタや、かのハンディ98も組み上げると取っ手のできる箱に入っていた。あれを買った人はみな重い思いをして持ち帰ったのだろう。ご苦労様。

*

無事帰宅して、取っ手に食い込んだ右手の指を剥がす。そして、2度3度右手を振って、感覚が戻るのを待たせよう。

やがて、自分の部屋へ重い箱を引きずっていき、わくわくしながら44,800円の箱を開ける。

箱の中に黒いボックスがあり、それごと、7冊のマニュアルと、黒いディスクケースを取り出す。箱はその辺にうっちゃっておき、マニュアルとディスクケースを眺める。

XC ver.1でばりばりやっていた貴方、きっと苦もなくver.2.0の世界へ行けたでしょう。けれど、初めてXCというものに触れた方、気の迷いで買ってしまった方。7冊のマニュアルのどれを見ても「導入マニュアル」だとか「はじめに読む本」なんてものではない。うーん。

答えは、「ユーザーズマニュアル」である。第1章に「お使いになる前に」とある。なかなかの不親切さ加減で、インストールについて書いてある。

XC ver.2.0のシステムディスクは2枚。けれど、そのままでは使えない。そこでインストールという作業が必要となる。必要なXCのファイル(パッケージ)を、使用するHuman68kのシステムに組み込んでやることだ。フロッピーベースだと実行用ディスクを1枚にまとめないと不便だからね。XC ver.2.0には、こういった実行用ディスクの作成や、ハードディスクにインストールするためのインストーラ“INSTALL.BAT”がついている。

私はインストーラを使わず、手作業でファイルを移した。理由はまあインストーラ

に対する不信感とか、添付のインストーラでインストールできるだけのハードディスクの空き容量がなかったことなどなどだ。

インストーラを使うにしても、手作業でコピーするにしても、必要なファイルとそうでないファイル、さらにはどのファイルはどんな役目を背負って生まれてきたかわからないとドンキホーテだ。無謀だ。

初めてCコンパイラを手にする人は、そのファイルの多さにとまどう。はじめてのCやK&Rをどれだけ読んでも、そんな話は書いていない。プログラム開発前にこういったファイル構成や、各々の役割を知っておくと非常にあとが楽だ。

よって、以下の章は、XCワールドを歩きまわるためのガイドマップである。

システムディスク1を歩く

図1-1がシステムディスク1のディレクトリである。見てわかるとおり、ディレクトリは5つである。一見、Cに関係するファイルはないかの如く振る舞っているが、あにはからんや、そんなことはない。

XCとは何か。ただのCコンパイラではなく、X68000の開発環境セットなのである。

今のところXCについているのが最新バージョンだと思えるので、

“自分のX68000の環境を最新のものにリプレースする”

ことも考える必要がある。

特に、Human68kのバージョン1を使っていた人は、これを機にバージョン2にしようのがよい。バージョン2の人も、今回のver.2.02であるから、新しくしてしまおう。特に、SYSディレクトリにあるもの、BINディレクトリにあるものはいつの間にかバージョンアップしてたりするので、自分が使っているディスクに全部入れてしまおう。それが嫌な人は、ディレクトリを取り、日付の新しくなったものだけをコピーすればいい。COPYALLコマ

ンドをうまく使えば、簡単にできる。なにげなく日付が新しくなっているものもある(図1-3)。そうでなくても、知らぬうちに自分のより新しいものがあつたりするものだ。

さて、この中で何が必要であるか。

図1-2にシステムディスク1の全ファイルを示した。“おニュー”とあるのは、SX-WINDOW以降に追加されたものだ。今回のXCで初めてついたものもある。SUPER-HDについていたものもある。

“新ver.”とあるのは、とりあえず、なにげなくバージョンアップしていただものだ。日付が'90年のものにつけておいた。Human68kがSCSI対応になったことに対応するバージョンアップが多いと思われるが、そうでなくても、隠れて日付が変わってたりするので要注意だ。

“ver.確認”とあるのは、こういうファイルはまず疑ってみなさい、ってことだ。

1) SYSディレクトリ

あやしいのが“SCSIDRV.SYS”と“OPMDRV2.X”。前者がSCSIドライバってことはわかる。後者はあやしい。実はこれ、MIDI対応のOPMドライバなのだ。MIDIを持ってない人、持っていない間に合っている人は使う必要がないので、気にしなくてもいい。

気にすべきは、“IOCS.X”だな。これを使うと、テキスト表示やグラフィックが速くなるというものだ。使うも使わないも好き好きだが、私はCONFIG.SYSに組み込んで使っている。これを使うと、MZ-2500ばりのスムーズスクロールも体験できる。

2) BINディレクトリ

こいつもあやしい。“*”のついているのが、とりあえず開発必需品(あるいは準必需品)だ。このうち、なにがなんでも必要なのが“LK.X”，つまりリンカだな。こいつは新バージョンでないとコンパイルできない。あとのは、コンパイルするだけならなくてもいい。コンパイルするだけ、

ならね。

まあ、いろいろと便利なものが多いので、全部インストールしてしまおう。

3) その他

とりあえず、ETCディレクトリはいらない。インストールしてしまえば用済みだ。

システムディスク2を歩く

いよいよコンパイラ本体の登場だ。図2-1は図1-1と同様、システムディスク2の中のディレクトリである。これだけある。SAMPLE以外は必要そうなものばかりだ。

いきなり、図2-2の全ファイル図を覗いて、そのガイドを始めることにする。

1) CCディレクトリ

C Compilerってわけで、CCである。かつて、このディレクトリにはCC0とかC C1とかCCPとかいろんな子供がいたが、今回の新バージョンでは“CC.X”ひとつしかない。普通に考えて、“前のバージョンでいろいろ分かれていたプログラムを1

図1-1 システムディスク1のディレクトリ

```
¥
|
|-SYS
|
|-BIN
|
|-ASK
|
|-HIS
|
|-ETC
```

図2-1 システムディスクのディレクトリ

```
¥
|
|-CC
|
|-INCLUDE
|
|-LIB
|
|-BASIC2
|
|-BC
|
|-SAMPLE
```

図1-3 SYSディレクトリの内容

XCシステム#1 14 ファイル ファイル使用量	B:¥SYS 972K Byte 使用中 238K Byte 使用	249K Byte 使用可能
PRNDRV	SYS 1816	89-02-10 12:00:00
PCMDRV	SYS 416	87-05-15 12:00:00
ASK68K	SYS 121470	89-04-04 12:00:00
RAMDISK	SYS 1816	89-02-10 12:00:00
SRAMDISK	SYS 924	87-05-15 12:00:00
PRNDRV1	SYS 3566	87-05-15 12:00:00
PRNDRV2	SYS 1816	87-05-15 12:00:00
PRNDRV3	SYS 1816	87-05-15 12:00:00
SCSIDRV	SYS 1274	90-06-15 12:00:00
OPMDRV2	X 35798	90-05-05 12:00:00
FLOAT1	X 11498	87-11-03 12:00:00
FLOAT2	X 12844	90-05-05 12:00:00
HISTORY	X 27830	89-04-04 12:00:00
IOCS	X 14420	90-06-15 12:00:00

←いつの間にか新しくなった

図1-2 システムディスク1の全ファイル

```
¥
|-XCシステム#1
|-HUMAN.SYS
|-CONFIG.SYS
|-KEY.SYS
|-USKCG.SYS
|-BEEP.SYS
|-STARTUP.ENV
|-AUTOEXEC.BAT
|-COMMAND.X
|-SYS
|-PRNDRV.SYS
|-PCMDRV.SYS
|-ASK68K.SYS
|-RAMDISK.SYS
|-SRAMDISK.SYS
|-PRNDRV1.SYS
|-PRNDRV2.SYS
|-PRNDRV3.SYS
|-SCSIDRV.SYS
|-OPMDRV2.X
|-FLOAT1.X
|-FLOAT2.X
|-HISTORY.X
|-IOCS.X
|-BIN
*|-AS.X
*|-LK.X
*|-DB.X
*|-SCD.X
*|-SCD.CNF
*|-SCD.HLP
|-AR.X
*|-LIB.X
*|-CV.X
|-BIND.X
|-CASE.X
|-DRIVE.X
|-MOVE.X
|-PROCESS.X
|-COPYALL.X
|-TERM.X
|-TOUCH.X
|-TREE.X
|-WHERE.X
|-FORMAT.X
|-DISKCOPY.X
*|-MAKE.X
|-PRINT.X
*|-ED.X
*|-ED.HLP
|-ASK
|-ENV1.ASK
|-ENV2.ASK
|-ENV3.ASK
|-ENV4.ASK
|-ENV5.ASK
|-HIS
|-KEY.HIS
|-HISTORY.HIS
|-HISTORY.HLP
|-ETC
|-INSTALL.BAT
|-FD.BAT
|-FDR.BAT
|-HD.BAT
|-YN
|-TOOL.X
```

←ディスクの名前

←Ver.確認

←環境変数をチェック
←Ver.確認

←おニュー

←おニュー

←新Ver.

←おニュー

←新Ver.

←新Ver.

←新Ver.

←おニュー

←おニュー

←おニュー

←新Ver.

←新Ver.

←新Ver.

←新Ver.

←新Ver.

←新Ver.

←おニュー

←新Ver.

←新Ver.

←インストーラ

←インストーラで使用

←//

←//

←//

←//

図2-2 システムディスク2の全ファイル

```
¥
|-XCシステム#2
|-CC
|-CC.X
|-INCLUDE
|-ASSERT.H
|-AUDIO.H
|-BASIC.H
|-BASIC0.H
|-CLASS.H
|-CONIO.H
|-CTYPE.H
|-DIRECT.H
|-DOSLIB.H
|-ERROR.H
|-FCNTL.H
|-FCTYPE.H
|-FLOAT.H
|-GRAPH.H
|-IMAGE.H
|-IO.H
|-IOCSLIB.H
|-JFCTYPE.H
|-JSTRING.H
|-LIMITS.H
|-MATH.H
|-MOUSE.H
|-MUSIC.H
|-MUSIC2.H
|-PROCESS.H
|-SETJMP.H
|-SIGNAL.H
|-SPRITE.H
|-STAT.H
|-STDARG.H
|-STDDEF.H
|-STDIO.H
|-STDLIB.H
|-STICK.H
|-STRING.H
|-TIME.H
|-TIMEB.H
|-UTIME.H
|-DOSCALL.MAC
|-ERROR.MAC
|-FCNTL.MAC
|-FDEF.H
|-FEFUNG.H
|-IOCSCALL.MAC
|-LIMITS.MAC
|-MALLOC.MAC
|-MATH.MAC
|-PROCESS.MAC
|-STAT.MAC
|-STDIO.MAC
|-TIME.MAC
|-LIB
|-CLIB.L
|-BASLIB.L
|-DOSLIB.L
|-IOCSLIB.L
|-FLOATFNC.L
|-FLOATEML.L
|-BASIC2
|-BASIC.X
|-BASIC.CNF
|-MUSIC2.FNC
|-AUDIO.FNC
|-GRAPH.FNC
|-MOUSE.FNC
|-STICK.FNC
|-IMAGE.FNC
|-SPRITE.FNC
|-BC
|-BC.X
|-AUDIO.DEF
|-MOUSE.DEF
|-SPRITE.DEF
|-STICK.DEF
|-IMAGE.DEF
|-GRAPH.DEF
|-BASIC.DEF
|-MUSIC2.DEF
|-BASIC.CNF
|-SAMPLE
|-SAMPLE.DOC
|-MESH.C
|-DEF.BAS
|-DEF.HLP
|-MAKEFILE
```

←ディスクの名前

←コンパイラ

←インクルードファイル
が入っている

←ライブラリ

←BASIC

←新しいVer.

←おニュー

←BASIC→Cコンバータ

←おニュー

←サンプルプログラム

本にしたな”って結論が導き出される。そのとおりで、CC.Xも400Kバイト以上と、破格の大きさだ。とっても大きい。おかげでメインメモリ1Mバイトユーザーは泣く泣く2Mバイトに増設するか、泣く泣く日本語FEPなどを削って最小限の環境で使うか、どっちにしても泣くしかなかった。「Cコンパイラを買うくらいのユーザーは、きつと、メモリの増設くらいしてるだろう」って思ったのかもしれない。

CC.Xに統合化された機能は次のとおり。
CC.X 以下のプログラムをコントロールするメインプログラム。

BC.X BASICプログラムをCプログラムに変換するためのもの。

CCP.X PはプリプロセッサのP。プリプロセッサってのはプリプロセスを行うプログラムのことで、プリ(PRE)は“あらかじめ”でプロセスは“処理”だから、ここでは“コンパイルの前にあらかじめ行う処理をするプログラム”って意味だ。

後述のインクルードファイルのところで、こいつの役割を解説しよう。

CC0.X, CC1.X CCPが解析したプログラムをアセンブラのソースプログラムに変換する、コンパイラの本体部分。

CC2.X オプティマイザといわれる。CC1.Xの吐き出した出力を最適化したアセンブラのプログラムに変換するものだが、たいしたことはしてくれなかったらしい。

AS.X アセンブラである。アセンブラのソースプログラムをアセンブルするプログラム(ああ、わかんない人が読むと何かなんだかわかんない文章だ)である。AS.Xはシステムディスク1に入っていたではないか、ってな話もあるが、あれは、アセンブラで開発する人のためである。

と、この7本がひとつになったのだから、でかくて当たり前かもしれない。これによって、全体のコンパイルに要する時間は短くなったらしい。が、CC.Xを読む時間が長くなったので、えんえんとCC.Xを読んだ方がいいがエラーがたくさん出てあっさりと終わり、なんてときは腹が立つぞ。

2) INCLUDEディレクトリ

インクルードファイルが入っているディレクトリだ。includeというのは“(全体の一部として)含める”っていう意味。

includeディレクトリには拡張子が“.H”のファイル41個と、拡張子が“.MAC”のファイル11個が入っている。

Hってのは“ヘッダ”，MACってのは“マクロ”の略号である。まず，Cに馴染み深いヘッダファイルから見ていこう。

図3 MATH.Hの中身

```

/*
 * math.h X68k XC Compiler v2.00 Copyright 1990 SHARP/Hudson
 */
#ifdef FORWARD
#define __PROTO_TYPE
#endif
#ifdef __STDC__
#define __PROTO_TYPE
#endif

/**** math exceptions ****/
struct exception {
    int     type;
    char    *name;
    double  arg1, arg2;
    double  retval;
};

struct complex {
    double  x,y;
};

/** Exception type codes **/
#define DOMAIN      1
#define SING        2
#define OVERFLOW    3
#define UNDERFLOW  4
#define TLOSS       5
#define FLOSS       6

#define PI          3.1415926535898
#define PID2        1.5707963267949
#define PID4        0.78539816339745
#define I_PI        0.31830988618379
#define I_PID2      0.63661977236758
#define HUGE_VAL    3.5953862697247E+308
#define HUGE_VAL    1.1125369292536E-308

/*
 * extern volatile int  errno;
 */
#ifdef __PROTO_TYPE

double  acos(double);
double  asin(double);
double  atan(double);
double  atan2(double, double);
double  cos(double);
double  sin(double);
double  tan(double);
double  cosh(double);
double  sinh(double);
double  tanh(double);
double  exp(double);
double  frexp(double, int *);
double  ldexp(double, int);
double  log(double);
double  log10(double);
double  modf(double, double *);
double  pow(double, double);
double  sqrt(double);
double  ceil(double);
double  fabs(double);
double  floor(double);
double  fmod(double, double);
double  hypot(double, double);
double  except(int, char *, double, double, double);
int     matherr(struct exception *);
double  cabs(struct complex *);

#else

double  acos();
double  asin();
double  atan();
double  atan2();
double  cos();
double  sin();
double  tan();
double  cosh();
double  sinh();
double  tanh();
double  exp();
double  frexp();
double  ldexp();
double  log();
double  log10();
double  modf();
double  pow();
double  sqrt();
double  ceil();
double  fabs();
double  floor();
double  fmod();
double  hypot();
double  except();
int     matherr();
double  cabs();

#endif

```


うんちゃら.HってファイルはすべてCのインクルードファイルである。

たとえば、以下のプログラムがあったとする。cos(π)の値を求めている例だ(ただし、このままではなんの役にも立たない。だって、結果をどこにも出力していないんだもん)。

例1)

```
#include <MATH.H>
double K;
main()
{
    K=cos(PI);
}
```

この一番上の#includeってのがインクルードファイルを呼び出すプリプロセッサコマンドである。頭に#がついているとプリプロセッサコマンドだよ、っていう印になり、includeってのは、次のファイル名のファイルをコンパイルの前にこの位置にくっつけてちょ！てな意味となる。こいつはプリプロセッサの仕事で、CC.Xの中に含まれている。

さて、では、このMATH.Hはどうなっているか。それが図3である。中身はテキストファイルであり、こうなっているのだ。中に#うんちゃらっていうコマンドが山ほどあるが、それは後ろで中森氏が説明してくれるはずだ。この中で、MATH、つまり数学関数で使う定数や関数の名前や構造の定義なんかをしているのがわかるはずだ。C自体は“PI”なんていう定数は持っていないのだが、ここで定義しているので、#includeしておけば、勝手にプログラム中で使ってかまわないのである。

さて、こいつがプリプロセッサの段階でプログラムにくっつくわけであるが、実のと

ころ、インクルードファイルの中の#うんちゃらもプリプロセッサコマンドなのでプリプロセッサに解釈され、その結果だけがプログラムにくっつく。

具体例を見てみよう。

先の例1をプリプロセッサにかけた結果が図4だ。こいつはCC.Xの/Pオプションで見ることができる。#うんちゃらがなくなったのがわかってもらえるだろうか。数学的関数の名前がずらっと宣言されたのがわかるだろうか。

そんでもって、#defineってやつで図3で定義されていたPI(π の値)が実際の数字に置き換わったのがわかるだろうか。

プリプロセッサはこういう仕事をするのであり、プリプロセッサコマンドはこう使うのである。

インクルードファイルはこのように単純なテキストファイルなので、自分で勝手に“OGIKUBO.H”なんてファイルを作って、中で勝手に、

```
#define OGIKUBO 1024
K = OGIKUBO * 2;
```

って書くだけで、Kに2048っていう値が求められるのだ。なんて具体的な説明だ。

ここでちょいと図4に戻る。中にある、“# 数字 ファイル名”

の謎だ。この数字が何を表すコードかってのはさておいて、MATH.Hっていうファイル名がフルパスで書かれている。どうしてフルパスで書かれているのか。これはプリプロセッサがいろんなディレクトリを探してやっとこさMATH.Hを探し当てたっていう意味ではない。

あらかじめ、Human68kの環境変数ってやつに、

include=D:¥XC¥INCLUDE

ってのが定義してあったからだ。プリプロセッサはインクルードファイルを探すとき、includeっていう環境変数を参照するのだ。だから、includeって環境変数名に割り当てたディレクトリか、カレントディレクトリか、コンパイル時のオプションで指定したディレクトリにうんちゃら.Hを置いておく必要があるのだ。これが、図1-2の“環境変数をチェック”の意味だったりする。

INCLUDEディレクトリの話からプリプロセスの話まで流れてしまったが、まあ、避けて通れないこともないが、わかっているとなおおいしいっていう道なので、お得なのだ。

ちょっとだけ付け加えておく。MACファイルの話だ。こいつはアセンブラで開発する人以外はきっと知らないで、気にしないようにしよう。アセンブラで使う汎用マクロファイルが入っているのだ。

3) LIBディレクトリ

CC.XはCで書かれたプログラムを最終的にアセンブルまでする。ver.1ではアセンブラのプログラム(“うんちゃら.S”ってやつ)を出力したけど、ver.2ではそれがない。直接、アセンブル後の“うんちゃら.O”というオブジェクトプログラムが出力される。

さて、INCLUDEのところで書いたように、#includeで読み込んだ“うんちゃら.H”ファイルには、関数の宣言が入っていた。というより、宣言しか入っていない。その関数の本体、つまり実際に処理をする部分はそこにあらず、なのである。さすがに宣言だけすればいいというわけにもいかない。そんなにこの世は甘くない。

図4 プリプロセッサのお仕事例

```
# 1 "fig7.c"
# 1 "D:¥XC¥include¥math.h"
# 6 "D:¥XC¥include¥math.h"

struct exception {
    int type;
    char *name;
    double arg1, arg2;
    double retval;
};

struct complex {
    double x,y;
};

extern volatile int errno;

double acos(double);
double asin(double);
double atan(double);
double atan2(double, double);
double cos(double);
double sin(double);

double tan(double);
double cosh(double);
double sinh(double);
double tanh(double);
double exp(double);
double frexp(double, int *);
double ldexp(double, int);
double log(double);
double log10(double);
double modf(double, double *);
double pow(double, double);
double sqrt(double);
double ceil(double);
double fabs(double);
double floor(double);
double fmod(double, double);
double hypot(double, double);
double except(int, char *, double, double, double);
int matherr(struct exception *);
double cabs(struct complex *);

# 102 "D:¥XC¥include¥math.h"
# 2 "fig7.c"
double K;
main()
{
    K=cos(3.1415926535898);
}
```


どこに入っているか。

結果として、必要な関数の機能を記述したプログラムが“うんちゃら.X”になった状態でその中に組み込まれていれば問題なく動作するわけである。実際、問題なく組み込まれることになっている。

それが、LIBディレクトリに入っている“なんちゃら.L”というライブラリだ。LIBというのはライブラリ (LIBRARY) なわけだね。

ここにはインクルードディレクトリとは違って、6つしか入っていない。名前を見ればわかるとおり、C専用関数のライブラリとBASIC TO C用のBASICと同じ関数のライブラリと、DOSコールの機能を関数にしたやつと、IOCSコールの機能を関数にしたやつと、浮動小数点演算関係のライブラリだ。これらにはいろんな関数のオブジェクトプログラムがかけ集められて、ぐしゃっと詰まっている。

CC.Xが吐き出したオブジェクトプログラムに、このライブラリから必要なものを組み込んでやらねばならないわけだ。

それがリンク (LK.X) の仕事である。どの関数を使うよん、ってのはこっそりと“うんちゃら.O”の中に書いてあるので、リンクさんはライブラリから必要なものだけを“うんちゃら.O”に組み込んで、単体で実行できる形式にしてやるのだ。この作業をリンクという。

そして、LK.Xが“なんちゃら.X”を吐き出して、無事、コンパイルは終了するわけだ。あとは、“なんちゃら”って、キーボードから打ち込んで、うまく動くのを確認すれば一件落着である。

図1-2のBINディレクトリにあった“LIB.X”ってのはユーザーが自分で“なんちゃら.LIB”を作ったり、“なんちゃら.LIB”に新しく追加したりするツールだ。ライブラリアンという。

includeがそうだったように、ライブラリも“ライブラリはここにあるよん”と教えてやる必要がある。そのための環境変数

がlibであり、

```
set lib=D:\XC\LIB
```

のように、AUTOEXEC.BATかどこかで指定しておく必要がある。

4) BASIC2

BASIC2ディレクトリ。BASIC.X、つまりBASICインタプリタが入っているディレクトリである。BASIC.Xは前の“DRIVE.X”コマンドによる論理ドライブの変更や、SUBSTコマンドによる仮想ドライブに対応していなかった”ボケが改善されたなどの変更があるようだ (つまり、Human68k ver.2対応になった)。

MUSIC2.FNCはMIDI関係の関数が追加になった。OPMDRV2.Xを組み込んで、MIDIを使う人が使う。そうでなくて、OPMDとかを使う人は使わない。

5) BC

BCディレクトリはBASIC TO Cコンバータ関係のディレクトリである。BASICで書かれたプログラムをCに変換するときに使い、BC.XにBASICプログラムを放り込むと、Cのプログラムが生成される。

BC.Xとともにずらっと並んだ“.DEF”ファイルはBASIC関数の定義ファイル。DEFというのは常々、DEFINEの略語である。図5はMUSIC.DEFの中身。このように、BASICで使用する関数の定義が入っていて、どの関数群を使うかはBASICインタプリタと同様、“BASIC.CNF”に書いてある。CNFってのはCONFIGURATIONの略だ。

CC うんちゃら.BAS

ってのを実行すると、CC.Xは“うんちゃら.C”、さらに“うんちゃら.O”を作成し、LK.Xのお世話になって、“うんちゃら.X”を作るのだ。

CC.Xさんが“.DEF”はどこかいな、って探すのにも環境変数を使う。

```
set bc=D:\XC\BC
```

といった感じで、環境変数bcに設定したディレクトリがそうだ。ちゃんとセットしておくように。

6) SAMPLE

SAMPLE.DOCはまあ、このディレクトリの中身の使い方だな。MESH.Cはともかくとして、DEF.BASはBASICのサンプルでついてきたスプライトエディタだが、コンパイルすると速くなるので便利。

MAKEFILEってのは、MAKE.XっていうUNIXゆずりのMAKEコマンド用のサンプルだ。MAKE.Xは図1-2のBINのディレクトリに入っている。

ライブラリディスクを横目で眺める

3枚目がいらぬ人はいらぬというディスクである。これはなくともコンパイルはできる。図にするまでもなく、入っているファイルは2つだけだ。

CLIB.ARCとBASLIB.ARCである。

これは拡張子からわかるとおり、アーカイブファイルである。ARCはARCHIVE Sの略で、記録保管所という意味。いろんなファイルを1本にまとめつつも、中ではすぐ必要なファイルが取り出せるようにちゃんと並んでいるものを指す。

BASLIB.ARCをダンプしてみると (コントロールコードが入っているのでTYPEコマンドでは見られないが)、BASIC関係のCの関数のアセンブラのソースが全部詰まっているのが見える。それを読んで勉強しても、必要なものだけ取り出してアセンブラプログラムを作るときに活用してもいい。

DOSLIB.ARCとIOCSLIB.ARCがないなどといわないように。あれらはそのままIOCSやDOSのルーチンをコールしているだけなので、アセンブラのソースをわざわざ載せることもあるまい。

XCへの道

これで、XCへの道は開かれたわけである。これからCに挑戦しようという皆様方、“べつにCでなくてもいいじゃん”てなことでもCを使ってやってください。

暇潰しプログラミング程度の短いやつでも、1回作ってしまえばいつでもコマンドシェルから呼び出せるっていうおいしさはBASICではなかなか味わえない。

あと、注意すべきは、“資源”ですな。ハードディスクは必需品。ハードディスクがなくてもRAMが4~6Mバイトくらい載っていれば大きなRAMディスクを確保できるからいいけどね。

というわけで、原の落球を見ながら (うふうふ) この原稿は終わる。

図5 MUSIC.DEFの中身

```
I      m_alloc(C,I)      :      (%,%)
I      m_assign(C,C)     :      (%,%)
I      m_cont(C,-,C,-,C,-,C,-,C,-,C,-) :      (%,%,%,%,%,%,%,%)
I      m_free(C)         :      (%)
I      m_init()          :      ( )
I      m_play(C,-,C,-,C,-,C,-,C,-,C,-) :      (%,%,%,%,%,%,%,%)
I      m_stat(C)         :      (%)
I      m_stop(C,-,C,-,C,-,C,-,C,-,C,-) :      (%,%,%,%,%,%,%,%)
I      m_tempo(C)        :      (%)
I      m_trk(C,S)        :      (%,%)
I      m_vget(C,CA)      :      (%,%)
I      m_vset(C,CA)      :      (%,%)
```


Cコンパイラが取り扱うファイル

ひとりのプログラマがCでプログラムを作ろうと思いついてからそのプログラムが完成するまで、いろいろなファイルが作成され、参照され、そして生成される。順番に見ていこう。

まず、以下の説明でよくよく顔を出す「スイッチ」という用語は、コンパイルの条件を設定するためにプログラマがコマンドラインから指定する文字列で、コンパイルオプションとも呼ばれる。たとえば、

```
CC /W /Y myprog.c
```

としてコンパイルするとき、Cコンパイラに/Wおよび/Yというスイッチを与えたことになる。スイッチはスラッシュ(除算記号「/」)やハイフン(減算記号「-」)のあとに英数字をくっつけた形で与える習慣になっている。XCはどちらも認識してくれるようだが、GCC (GNU C) はハイフンしかスイッチとして認識しない。

*

●～.c (作成)

いわずと知れたCのソースファイル。基本的には、プログラマはこのファイルだけをED、Xなどのテキストエディタで書いてコンパイラにかければよい。

●～.h (作成または参照)

ヘッダ(header)ファイルの略。プログラムの先頭には、

```
#include <stdio.h>
```

```
#include "myheader.h"
```

などを書いてヘッダファイルを指定するのが習慣になっているからである。

ヘッダの役割は、主に、ライブラリに格納されている関数の宣言、ライブラリ関数で用いる定数や構造体などの宣言、マクロの定義などである。通常ヘッダファイルには関数の本体(プログラム)は書かない。関数本体は、あらかじめコンパイルしてライブラリ化し、ユーザープログラムをコンパイルする際にリンクするほうが、コンパイル時間が短くなるからだ。

プリプロセッサ命令#includeのあとに、ヘッダファイル名を、

```
<(ファイル名)>
```

と不等号で囲むと、環境変数includeで指定されたパス(システム標準のヘッダファイルを格納しているディレクトリ)だけから指定されたヘッダファイルを探しに行く。

```
"(ファイル名)"
```

のようにダブルクォートで囲んだ場合、環境変数includeが指すディレクトリの前に、カレントディレクトリも調べることになっている。

たとえば、<>はシステムが付けてくれる標準のヘッダファイルを指定するのに用い、対して自前で作ったヘッダファイルはカレントディレクトリに置いて、" "で指定するといった使い分けをするときに都合がいい。

●～.p (生成)

XCで/Pスイッチをつけたときに生成されるプリプロセッサの出力で、通常のコンパイルで表に現れることはない。プリプロセッサとは、ソースファイルがCコンパイラ本体で処理できるように加工するプログラム。具体的にはインクルードファイルやマクロの指定を展開する。コンパイルの前処理(プリプロセッサ)を行うという意味でプリプロセッサと呼ぶ。

GCCでは-Eスイッチでプリプロセッサの出力が標準出力(つまり画面)に出てくる。ファイルに保存したいときはリダイレクションを用いるとよい。

●～.s (作成または生成)

多くのCコンパイラは、直接オブジェクトを出力することはない。Cコンパイラの仕事は、パーサやコードジェネレータを呼び出し、ソースファイルを解釈してアセンブリリストを出力することであり、アセンブラを呼び出してオブジェクトファイルを生成させることである。コンパイラが出力するアセンブラのソースファイルがこの～.sファイルだ。

ただし、GCCでは実行時に特にスイッチ(-S)をつけない限り、生成されない。アセンブラのソースは、実行に必要なないのだ。余分なファイルは要求されない限り残さないというのがGCCの作法であるように思われる。

●～.mac (参照または作成)

アセンブルする段階で、iocsコールやDOSコールを使うことは多い。その関連のマクロを集めたのが、ヘッダと同じく環境変数includeのディレクトリに入っている、

```
iocscall.mac
```

```
doscall.mac
```

である。なお、上で書いたように～.sを自分で作るなどとは、このマクロ定義ファイルを自分で書く可能性もある。

●～.o (生成)

アセンブラが出力するオブジェクトファイル。このままではまだ実行できない。最後のリンクフェーズを経て、ようやく一人前の実行可能ファイルになる。

GCCでは実行時に特にスイッチ(-O)をつけない限り、生成されない。このファイルも初めてのうちは必要ないだろう。自分でライブラリを作るようになれば、～.oファイルを作る必要も出てくる。

●～.a(.l) (参照または作成)

Cプログラミングにおける影の立役者、それがこのライブラリファイルである。

このライブラリファイルの正体は、たくさんのオブジェクトファイルである。Cで利用できる関数はこの中に収められている。リンクは、この中からユーザープログラムで参照されている関数(のオブジェクト)だけを切り出し、先ほどのオブジェクトプログラムと結合する。

Cのライブラリ関数はあまりにも数が多いので、それをいくつかのグループ分けし、ひとつのグループにつきひとつのファイルにまとめている。おかげで、ライブラリのディレクトリはすっきりしたものである。

バージョン1では、アーカイバというプログラムでまとめた～.aというファイルだったが、バージョン2ではライブラリアンというプログラムでライブラリ化し、拡張子も～.lである。したがって、旧バージョンのリンクでは、新しいライブラリを使った場合、素直にコンパイルできなくなっている。

インクルードファイルが格納されているのは環境変数includeが指すディレクトリであったが、ライブラリは環境変数libが指すディレクトリに格納されている。

XCでは、一部のライブラリ(lib.a)以外を用いるとき、/W, /Yスイッチが必要になっている。

ライブラリもユーザーが作ることができる。簡単に説明しておくと、まずCで関数を書く。このときmain()関数は書かない、つまり完結したプログラムにしない。それをコンパイルする。ただし実行ファイル(～.x)は作らずに、オブジェクトファイル(～.o)の段階で止めておく。スイッチはXCなら/Fc, GCCなら-c。それをアーカイバやライブラリアンで、ライブラリファイル、たとえばmyclib.aやmyclib.lにまとめる。

●～.x (生成)

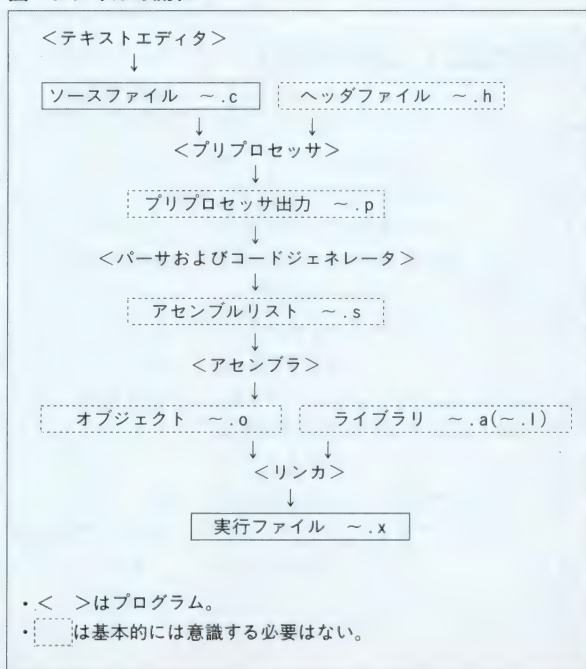
オブジェクトファイルとライブラリをリンクで結合して、ようやく出来上がった実行可能ファイルがこれ。もう、ふつうのDOS外部コマンドと同様に呼び出して使うことができる。

*

以上をまとめる。実行ファイルが出来上がるまでの流れは図のようになる。

プログラマはテキストエディタでソースファイルを書き、Cコンパイラを呼び出すだけだが、CC (GCC) ドライバは上に述べたようなたくさんのプログラムを次々と呼び出し、上に述べたようなファイルを生産するのである。

図 ファイルの流れ



XCを支援するおいしいツール

ソースコードデバッガを使ってみよう

Izumi Daisuke 泉 大介

C言語でプログラムを作成するにはまずED、Xなどのエディタでプログラムを書き、それをファイルに保存します。そのファイルをXCでマシン語に変換させ、やっと実行できるようになるのです。

しかも、通常は一発でマシン語プログラムへの変換が終了するわけではありません。人間ですからミスをします。修正するには再びエディタを使ってコンパイラが間違っていると指摘する行を表示し、プログラムを直さなければならないのです¹⁾。

エラーはまとめて出力され、最初は山のようなエラーメッセージと格闘することになるでしょう。エラーのなかでも特にセミコロンの付け忘れは、信じられないようなエラーメッセージをもたらします。図1はリスト1の4行目のセミコロンを取ってコンパイルしたときに表示されるエラーメッセージです(Ver.2.0にてコンパイル)。このように多くの場合セミコロンのない行の次の行番号が表示されます。しかも、セミコロンがないとは指摘してくれません²⁾。

止められない、止まらない

無事コンパイルが終了すれば、～.Xというマシン語ファイルが作成されます。が、ちゃんと動くという保証はどこにもありません。Cコンパイラは字面のチェックを行うだけで、実際に動かしたときにどうなるのかまでは面倒をみてくれないのです。

作成されたマシン語ファイルを動かして動作のチェックを行っていくのですが、ここで第2の関門が待っています。X-BASICならプログラムの実行中にBREAKし、変数の内容を表示してチェックできます。また、規定外の値を使用した場合には自動的にエラーで実行が中断されます。ところが生成されたマシン語ファイルの実行では途中で中断することはできませんし、規定外の値を使ったからといってプログラムが止まることもありません。

プログラムには、文字入力などBREAK

キーが効く場所も存在します。しかしこの場合、BREAKした途端にプログラムの実行は終了してしまいますので、BREAKして変数の内容を確認することはできません。変数の内容を調べたいと思ったら、怪しそうなところに変数を表示するプログラムを書き込み、再びコンパイルして実行する。これが従来行われてきたもっともオーソドックスなチェック方法です。さもなくば、デバッガ(DB、X)を使うしかありません。これは変換されたマシン語を対象としますので、マシン語の知識とCがどのようなマシン語を生成するかを知らなければにっちもさっちもいかないという代物です。

C言語で書いたプログラムを見ながら、自由に実行を中断したり変数を見ることができれば便利なのに……と、誰もが思うことでしょう。

ソースコードデバッガSCD.X

XCではVer.2.0からSCD.Xというソースコードデバッガが標準で付属しています。ソースコードデバッガを使うには、専用の情報をコンパイル後のマシン語ファイルが持っていなければなりません。Ver.2.0のXCでコンパイルするときに、

```
cc /Ns test.c
```

と「/Ns」スイッチを指定すれば、SCD.X用の情報をマシン語ファイルに持たせることができます。生成されたマシン語ファイルはそのままで実行可能ですので、プログラム作成中はこのスイッチを指定しておくことにしましょう。ではリスト1でSCD.Xを試してみることにしましょう。

●プログラムのコンパイル

ED、Xなどを使ってlist.cを入力したら、

```
A>cc /Ns list1.c
```

としてコンパイルしてください。

X68k XC Compiler ……

と表示されてコンパイルが始まります。間違いがなければそのまま何事もなく、

```
X68k Linker ……
```

と表示され、最後に「A>」の状態に戻ればコンパイル終了です。

不幸にしてエラーが出たときには、

```
A>cc /Ns list1.c > err
```

として実行してみてください。今度は画面にエラーメッセージを表示する代わりにerrというファイルが作成され、この中にエラーメッセージが収められます。そこで、

```
A>ed err
```

として、このファイルをED.Xに読み込みます。最初のエラーメッセージでカーソルが点滅していますね。ここで「ESC」「V」の順にキーを押すと、あら、不思議。エラーが出たファイルを自動的に読み込み、該当行にカーソルが飛んでいきます。もう一度エラーを見たいときには「ESC」「A」と入力します。以後「ESC」「A」

1) 最近のCコンパイラには、エディタ、コンパイラ、そしてあとで説明するソースコードデバッガを統合したものが存在します。コンパイル時にエラーが発生すると該当行の該当箇所にカーソルが飛んでいき、すぐさま修正できるという環境が実現されているのです。手慣れたエディタではないという不満はあるものの、この便利さはうらやましいものです。
2) 最近になってようやく一部のパソコン用Cコンパイラでセミコロンがないというメッセージが表示されるようになってきました。

図1 エラーメッセージの例

```
list1.c 5 :Error 22:declaration error.
list1.c 5 :Error 56:compound statement error.
list1.c 6 :Error 9:external definition error.
list1.c 10 :Error 61:operand type mismatch.
```

リスト1 コマンドラインの引数を表示する

```
1: #include <stdio.h>
2:
3: void main( argc, argv )
4: int   argc
5: char  *argv[];
6: {
7:     int i;
8:
9:     for ( i=0; i<argc; i++ )
10:        printf( "%d:%s\n", i, argv[ i ] );
11: }
```


と押すたびにエラーメッセージとlist1.cが切り替わって表示されます。

プログラムの修正がすんだら再びコンパイル。この作業をエラーが出なくなるまで繰り返します。

●さあ、SCD.Xの出番だ！

コンパイルが終了したらSCD.Xを使ってみましょう。フロッピーディスクで使っている方は、ここでシステムを起動したディスクを再びAドライブにセットし直してください。SCD.Xは起動ディスクのほうに入っています。

準備が完了したら、

```
A>scd list1.x
```

でソースコードデバッガの起動です。写真1のような画面が表示されたことと思います。

一番上にはメニューが並び、2番目のウィンドウにはCのプログラムが、3番目にはCPUの変数（レジスタと呼ばれている）が表示されています。4番目がコマンド投入ウィンドウと呼ばれているもので、上級者になるとマウスでプルダウンメニューを開く代わりにここでコマンドを入力して作業します（彼らにとってはそのほうが速いらしい）。一番下のなにも表示されていないところはユーザースクリーンと呼ばれ、実行するプログラムが使用する画面です。マウスでクリックするとバーが現れます。いずれのウィンドウもバーを上下方向にドラッグすることによって大きさを変更できるようになっています。

ではlist1.xを実行してみましょう。実行の前に画面下の黒い部分で左ボタンをクリックしてユーザースクリーンのタイトルを表示させ、1行分上にドラッグしておくといでしょう。SCDはプログラム実行中にユーザースクリーンとSCDの画面を切り替えますので、起動時の状態では画面がチラチラして見づらくなるからです。ユーザースクリーンのタイトルを移動しておくともこの切り替えは行われません。

ExecメニューからRunを選ぶとプログラムは通常のスピードで実行されます。これでは面白くありませんから、次のSlowを選んで実行開始です。プログラム中の1行が網掛け表示され、それが下へと動いていったのがわかりましたか？ SCDは網掛け表示で現在実行中の行を示すようになっているのです。実行が終了すると

```
program terminated normally
```

とコマンド投入ウィンドウに表示されます。プログラムは問題なく終了したという意味です。

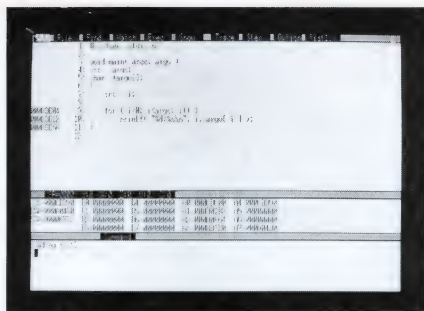


写真1

ソースコードデバッグの実際

リスト1のプログラムはコマンドラインで起動時に与えた引数を表示するものです。たとえば、

```
A>list1 a b c
```

と入力すると、

```
0 A: ¥list1.x
1 a
2 b
3 c
```

と画面に表示します。SCDでも引数を与えて実行させることが可能です。これはコマンド投入ウィンドウで行います。

```
-c a b c
```

と入力してください。最初の「-」はSCDが表示したプロンプトですから入力の必要はありません。こうしてからプログラムを実行すると上のようにコマンドラインで引数を与えたのと同じことになります。

●変数を表示する

今度はただ実行を黙って見ているだけでなく、変数がどのように変わっていくかを追いかけてみることにしましょう。リスト1で使っている変数の中から、

i: ループカウンタ

argc: 引数の個数

argv: 引数配列

の3つを見えます。変数を表示するにはWatchプルダウンメニューからSetを選択します（写真2）。以後この操作をWatch-Setのように略記します。変数名を入力するウィンドウが開きますから、まずiと入力してリターン。再びWatch-Setメニューでargcをセット。最後に「argv[i];s」とします。argvだけはi番目のものを文字列として表示することにしました（sで文字列として表示することを示す）。

プログラムは2度続けて実行することはできません。Exec-Restartメニューで実行前の状態に戻し、再びExec-Slowで実行してみてください。表示されている変数

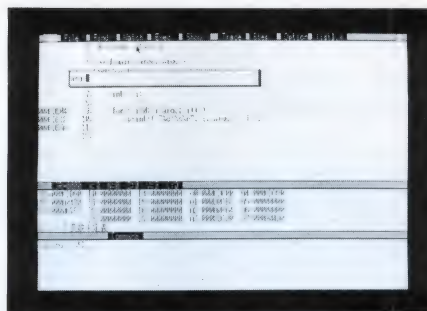


写真2

がどんどん変わっていきますね。

●1命令ずつ追いかける

Slow実行でもまだ速くて変数の変化についていけないかもしれません。こんなときには1命令ずつ実行するTraceメニューです。Exec-Restartで実行前の状態に戻したら、今度はTraceをクリックしてみましょう。画面の1行が網掛けになり、変数が変わりました。以後Traceをクリックするたびに1行ずつ実行されていきます。変数の変化を追うにはもってこいですね。

このプログラムを実行している間、ユーザースクリーンのほうはどう変化しているのでしょうか。画面を切り替えてみることにしましょう。まずメニューバーの右端にある□印のところをクリックし、SCDが全画面を占めるようにします。続いてShow-Screenメニューを選択するとユーザー画面が表示されます。ここでマウスのボタンかキーを押すと、元のSCDの画面に戻ります。先に説明したように、プログラム名とSCDのcコマンドで与えたパラメータが表示されていますね。ユーザースクリーンのタイトルバーを上スクロールさせた状態ではSCDの画面は見えません。□印をクリックするのを忘れないでください。

SCDでCの勉強を

リスト2は九九の表を画面に表示するプログラムです。今度はこのプログラムを実行してみることにしましょう。

```
A>cc /Ns list2.c
```

でコンパイルしたら、リターンキーを押して続けてカーソルを画面の最下行までもって

リスト2 掛け算のプログラム

```
1: #include <stdio.h>
2:
3: void main()
4: {
5:     int i, j;
6:
7:     for ( i=1; i<10; i++ ) {
8:         for ( j=1; j<10; j++ ) {
9:             printf( "%4d", i * j );
10:            printf( "\n" );
11:        }
12:    }
```

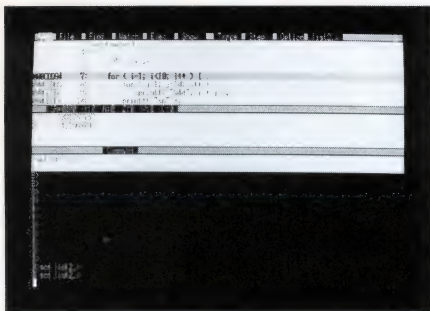



写真3

ってください。そこで、

A>scd list2.x
としてSCDを起動します。今度は画面の下半分をユーザースクリーンにして実行してみましょう。画面下の黒いところでマウスの左ボタンを押すと表示されるタイトルバーを、そのままドラッグして画面中央まで引き上げればOKです。レジスタ表示はとりあえず必要ありませんので、Show-Registerメニューを選択して消去します。表示する変数はiとjです。ではTraceで実行してみましょう(写真3)。

このプログラムではforの中でforが使っている、いわゆる2重ループになっています。Traceするとまず7行目が網掛け表示されiには1がセットされました。jには意味のない値が入っています。続いて8行目が網掛け表示され、jにも1がセットされます。そのあとは9行目から網掛け表示は動かず、変数jが1ずつ大きくなっていきます。ユーザースクリーンにはprintfを実行するたびに掛け算の答えが表示されていきますね。jが10になると内側のforの条件を満たさなくなり、網掛け表示は10行目に移ります。ここでユーザースクリーンは改行し、11行目が網掛け表示になったあと、再び8行目に戻ります。今度はiがひとつ大きくなって繰り返します。こうして1行ずつ実行していけばプログラムがどういう順序で動いていくのか一目了然です。途中までTraceしたら、あとはExec-Slowで一気に入行してかまいません。

プログラムの動きを目で見ながら、画面に表示されていく文字を追いかける。こんな芸当はX-BASICにもできません。他人の書いたプログラムを読むのはなによりの勉強だといわれますが、雑誌に掲載されたリストを見る際にはどの変数にどんな値が入っているのかを覚えながら見ていかなければならず、面倒なことこの上ない。SCDならこの作業を全自動でこなしてくれるのです。見たい変数があればWatch-Setで加えるだけ。実に簡単です。

「初心者の方にはリストを読むことをお勧めします」これまではこういわれてきました。しかしSCDが登場した今となっては、「初心者の方とはとにかく入力し、SCDで追いかけてみることをお勧めします」といったほうがいいでしょう。Cプログラミングも、SCDの登場によってプログラム実行時のチェックに関してはX-BASICを凌駕したといえます。

自作派に勧めるSCD

SCDはプログラムを1命令ずつ実行するだけでなく、目的の場所まで一気に実行することが可能です。実行を停止する場所には若干の制約があり、Cのプログラムの左に16進数が表示されている行しか指定できません。これは、この行に対してマシン語が生成されているという印で、Exec-SlowもTraceもよく見れば左に16進数を従えた行だけを実行していくのがわかると思います。ここではちょっと高度な使い方になりますが、自作プログラムをデバッグするのに有利な機能を紹介しておきましょう。

●関数をスキップしてTrace

Traceで1行ずつ実行すると、関数呼び出しがあった場合には呼び出された関数の中でも1行実行の対象となります。これに対し、呼び出された関数を高速実行するのがStepです。網掛け行は次の行へ移動しますので、現在注目している関数だけを1行実行しているという感じになります。デバッグが終わった関数までいちいち実行されるのは……、という場合にお勧めです。

●指定場所でプログラムを停止

任意の場所までプログラムを実行して停止させるのは簡単で、止めたい場所にマウスカーソルを移動し、右ボタンをクリックするだけです。簡単なプログラムを次第にバージョンアップしながらプログラミングを進めていく場合には、追加・変更した関数が呼び出されるまでは高速実行したいものです。このような場合に重宝する機能です。

Cでプログラムを作っていると、勘違いから結構アドレスエラーやバスエラーを引き起こしてしまうものです。ちょっと力のある人ならすぐさまDBXを起動し、バスエラーを起こした場所を突き止め、さらにpsコマンドでシンボルを表示させて、それがどの関数内で起きたのかを調べることでしょう。そのあとはどうしますか？ 再びエディタを起動し、プログラムとにらめっこというのが一般的なところでしょう。

SCDなら問題の関数まで一気に実行し、そこからTraceすることで簡単に問題のある行を見つけ出すことができます。そもそもこれらのエラーが発生した場所をShow-Assemblyメニューで逆アセンブルリストを表示して確認することができるので、psコマンドがどうしたこうしたという面倒もありません。逆アセンブルリストを逆スクロールさせていけば、関数名のシンボルを見つけ出すのは簡単です。

●プログラム停止の条件を指定

負の数を取らないはずの変数が-1になっているなど、変数が規定外の値を取っている。これもよくあるバグです。SCDはプログラムの実行を中止する条件を指定しておくことができます。Watch-WatchPtを選ぶと条件入力用のウィンドウが開き、停止条件をCの文法で記述します。たとえば

```
x <= 0
```

という具合です。条件が成立すればプログラムは停止します。

また、特定の範囲のメモリを監視することもできます。こちらはWatch-TracePtです。Cの配列は宣言した大きさを超えて書き込んでもエラーにはなりません。ところが大きさを超えた部分はほかの変数が使用している可能性がありますので、結果として誤動作してしまうことがあるのです。先の、変数の値がいつの間にか変わってしまうという原因が、範囲を超えたアクセスだったというのもよくある話です。

これら2つの機能はExec-Slow, Trace, Step実行中しか働きません。Exec-Runで高速に実行しながらチェックしてくれるとありがたいのですが、しかたがないので、変数が書き換えられてしまう関数呼び出しを見つけ出し、その関数の中をExec-Slowで実行してみるという使い方になるでしょう。

ソースコードデバッグは使わない。そう決めている、という人がいます。動いているのはCのプログラムではなくマシン語なのだから、マシン語用のデバッグのほうが便利だということです。そんな方に最後にひと言。SCDは従来のDBXを機能強化してソースコードデバッグに仕立てたものです。もちろんメモリダンプからレジスタの設定、アセンブル機能まで、あなたがデバッグに必要なと思っている機能はちゃんと用意されています。C言語レベルでデバッグできる範囲は見やすいソースコードで、もっと細かく調べたいときはマシン語レベルで、これひとつでデバッグ可能になっています。ほら、これならどうですか？

使って便利な外部コマンド

Human68kのシステムディスクにはBINというディレクトリがあり、さまざまな外部コマンドが入っています。これらは、いずれも～.Xの形の実行ファイルで、あとから次々と付け加えて、Human68kの機能を拡張することができます。

便利な外部コマンドは、Human68kのver. 2.0やXCでもいくつか追加されており、プログラム開発に積極的なユーザーのBINディレクトリはきつとかなり大きなものとなっていることと思います。ここでは、Human68kのver. 2.0やXCで付いてくる外部コマンドのなかから代表的なものを紹介しておきましょう。

なお、Human68k ver. 2.0で付いてくるものには (Hu)、XC ver. 2.0に付いてくるものには (XC) をつけました。また、特に表記のないかぎり、

コマンド名 スイッチ ファイル名 (ワイルドカード含む) として使います。

●CASE (XC)

指定したファイル (ただし読み出し専用およびシステム属性のある場合を除く) 名に含まれる半角の英文字を小文字 (/L)、または大文字 (/U) に変更します。また/Dでディレクトリ名を対象とします。ファイル名が省略されたときはカレントドライブ/ディレクトリのすべてのファイルが指定されます。

具体的にどのような場合に用いられるかというと、PC-9801へのファイル転送の際などです。MS-DOSではファイル名に半角英小文字を指定することができない (使うと大文字に変換してファイル名がつけられる) のでHuman68kで小文字でファイル名を付けられたものはそのままではファイル名を見ることはできても読み込むことができません。そこであらかじめHuman上で、

CASE A:

などとしてファイル名を半角大文字に変換するのです。

●DRIVE (Hu) (XC)

ドライブ名を交換します。またドライブ名を指定しない、あるいは1つだけ指定するとそのドライブの状態を示します。/Vオプションでボリュームラベルも見ることができます。

マシン語プログラムの開発中などでプログラムのなかにファイル名が組み込まれていてドライブを変更したいのにどうしてもドライブ名が変更できないことがあります。そのようなときに効果を発揮するのがDRIVEコマンドです。ファイル名を書き換えるのではなく疑似的にドライブを入れ換えることによって目標を達成することができます。

DRIVE A: B:

これでAドライブとBドライブを交換することができます。

●PROCESS (Hu) (XC)

現在使われているメモリが何に使われているかを知ることができます。また自分のマシンが

どこからどこまでにメモリが配置されているかを知っていればメモリの残り容量をだいたい知る (つまりMEMFREEを使わなくてもメモリの空き容量がわかる) こともできます。

また、Humanではバージョン2.0からバックグラウンドでタスクを実行するためのファンクションコールがサポートされた (今までのコマンドがバックグラウンドで動くわけではない) ので/bオプションでバックグラウンドプロセスの情報の表示、/aで従来のPROCESSの機能に加えバックグラウンドプロセスの情報を加えて表示を行います。

表示される内容は、

開始	終了	長さ	モード	ファイル名
006800	077FFF	071800	SUPER	Human.sys
:				
0F3480	0F367F	000200	MALLOC	
0F3690	17FFFF	08C970	USER	A:¥BIN¥PROCESS.x

で、

開始: メモリブロックの開始アドレス

終了: メモリブロックの終了アドレス

長さ: メモリブロックの大きさ

がそれぞれ16進数で表示されます。またモードのSUPER/USERはそれぞれプログラムがスーパーバイザー・ユーザーモードで実行されていることを示し、MALLOCはその表示のされている直前のプロセスによって確保されているメモリを示します。直前に表示されたファイル名+MALLOCがそのプロセスが実際に使用しているメモリの大きさということになります。そしてファイル名ですがプロセスの起動されたときのパス、ファイル名です。

●WHERE (Hu) (XC)

ファイルの検索を行うコマンドでデフォルトではカレントドライブから指定したファイルのパス名を表示しますがスイッチとして/A (すべてのドライブから探してくる) /F (パス名とファイル名を表示する) を指定することもできます。

Human68kやMS-DOSなどではファイル管理の方法として階層化ディレクトリを採用しており、ディレクトリを作りいくつかのファイルをそこに格納するのが普通です。この方法ではファイルの整理がしやすくなる反面、あまり複雑に階層をつけてしまうとファイルを格納してあまり長い時間がたつと忘れてしまうこともままあります。

たとえば、ゲームの原稿をどこにしまったのか忘れてしまった、たしかファイル名はGAMEという名前であったのだが……などという場合、

A>WHERE /a /f GAME.*

としてやれば、

a: ¥artic ¥game.10a

a: ¥artic ¥game.10b

c: は準備ができていません

(Cドライブにはフロッピーを入れていなかった) という具合に表示されます。

●TREE (Hu) (XC)

ディレクトリ構造を表示します。

特定のファイルを探すのではなく、どこに何があるのかドライブ全体の情報を知りたいときにファイル名の一覧表を表示させることができます。WHEREコマンド同様/スイッチがあり、すべてのファイル名を表示することができます (残念ながら/aスイッチはありません)。

また、TREEコマンドではすべてのファイルを表示するので特にハードディスクやMOなどを使用するとそのファイルの数は膨大なものになります。そこでファイル表示を見やすくするためにリダイレクションして、

A>TREE /f >ファイル名

A>ED ファイル名

としてエディタを使うか、あるいは、

A>TREE /f | MORE

とパイプして見るのが普通です。

●TERM (XC)

通信には通信ソフトを使うのが一般的ですが、RS-232C経由でコマンドのやり取りを行うだけであれば、実は通信用ソフトを使わなくてもとりあえず読み書きを行うことができます。XCに付属のTERMコマンドを使う方法がそれです。マニュアルには、

TERM……ターミナルモードに入ります。

と実に簡潔に書かれています。

A>TERM ログファイル名

とすることでRS-232Cとの通信が始まります。これでヘイズモデムがつながっているのならば、ATコマンドを使ってそのまま通信ができるわけです。

[F7] キーで表示内容のプリントアウト

[F8] で表示内容の記録

[F9] でコマンドモードへ

[F10] で終了します。

●MAKE (XC)

ファイル保守ユーティリティで、BINディレクトリに入っていますが、外部コマンドの規模を越えた多機能なツールです。これについては中森氏の記事をご覧ください。

*

最後に外部コマンドを使うコツなど。再帰的なようですが外部コマンドを使いこなすためのコツはやはりいろいろな局面でコマンドを使ってみるのではないかと思います。

というのもこういったものをうまく使いこなせるかどうかの決め手は、マニュアルにある機能をまる暗記することにあるのではなく、これはどんな局面でどう使えばよいのかということをも身体で覚えることにあるからです。いくら本を読んで理屈がわかっていても実戦経験がなければ戦っても勝てないのと同じことです (もちろん両方とも同じくらい実戦経験があれば理屈を知っているほうが有利だろうが)。

皆さんもぜひともいろいろコマンドを使ってトライ&エラーを繰り返してコマンドを使うノウハウを手に入れてみてください。

(古村 聡)

貴方のプログラミングを支援する

縁の下のプロセッサ

Nakamori Akira 中森 章

プログラマが一定の書式にしたがって書いたものを解釈してコンパイルできるようにお膳立てしてくれるのがプリプロセッサです。うまく使えば自分のスタイルによって、プログラムを簡潔に、そして効果的に記述することができます。

PASCALやFORTRANをよく知っている人がC言語を始めたときに一瞬つまづくのがプリプロセッサの機能です。そう、ソースプログラムの中の#includeとか#defineといった#で始まるあのオマジナイです。

大学時代私は先輩からポータブルCコンパイラのソースプログラムを見せてもらった（なんでもそれがC言語の標準的なプログラムらしい）のですが、その中の#ifdefの山にすっかりまいてしまいました。全然意味がわからなかったのです。そしてプリプロセッサの大切さを実感したのです。現在でもプリプロセッサをC言語のおまけ的な機能としてしか紹介してない参考書も多いようですが、本当にC言語を使いこなせるようになるためにはプリプロセッサ機能をしっかりと理解しなければなりません。しかし、あのK&Rの教科書でさえプリプロセッサ機能は4.11節でほんの少しだけ説明されているだけです。その重要度をつい見落としてしまいがちです。

その結果が何かわけがわからないままプログラムの先頭には必ず、

```
#include <stdio.h>
```

を書く習慣に落ち着くのです。実はこんなオマジナイはほとんどの場合不要なのです。私も会社に入って先輩からそのことを教えられたとき、頭をトンカチで殴られたような衝撃を受けました（今までは17文字も余分にプログラムを書いていたのか）。C言

語を勉強しているみなさんも、いつの日かそのことを知って大人になっていくのです。C言語のプリプロセッサ機能を単なるオマジナイとして終わらせたのではオモシロクありません。そこで、今回はこのプリプロセッサ機能の基本的な部分を紹介することにしましょう。

■ プリプロセッサとは

CコンパイラはC言語で書かれたソースプログラムを読み込み、コンパイルしてアセンブリ言語のプログラムを出力します。これがCコンパイラの役割です¹⁾。このとき、コンパイルの段階はさらに大きく2つの部分に分かれています。それが前処理（プリプロセス）とコンパイルです。つまり、C言語で書かれたソースプログラムは前処理によっていったん加工され、その加工されたあとのプログラムが実際にコンパイルされるのです。

C言語のプログラムの中によく見受けられる#includeとか#defineといったオマジナイはこの前処理段階で解釈されソースプログラムの加工が行われます。そしてこの前処理を行うためのプログラムがプリプロセッサなのです。

多くのCコンパイラの処理系ではプリプロセッサとコンパイラ（の本体）は別のプログラムで供給されています。たとえば、

リスト1

```
1: /*
2:      プリプロセッサ機能（イントロダクション）
3:
4:      これはなんでしょう
5: */
6: #define I          main
7: #define X68000    {
8: #define best      "こんにちは、世界\n"
9: #define computer  printf
10: #define in        )
11: #define is        (
12: #define that      }
13: #define the
14: #define think     (
15: #define this      ;
16: #define world     }
17: /*
18:      これは単なる英文に見えますが……
19: */
20: I think that the X68000 computer is the best in this world
```

UNIXのCコンパイラではプリプロセッサはcpp、コンパイラはccomであり、XCのバージョン1ではプリプロセッサはCCP、X、コンパイラはCC0、XとCC1、Xになっています。このようにプリプロセッサとコンパイラが分かれている理由は、プリプロセッサの提供するマクロなどの機能が結構強力であるため、プリプロセッサを単独にファイル変換プログラムとして利用する目論みがあるのでしょう。ただし、RUN/CやTurboCなどコンパイル速度を「売り」にしているいくつかのコンパイラではプリプロセッサとコンパイラは一体になっているようです²⁾。

さて、プリプロセッサの役割について説明しましょう。プリプロセッサの機能を大まかにいうと

- 単語の置き換え
- 条件付きコンパイル
- 別ファイルの取り込み

の3つです。このほかにも細々とした機能はあるのですが、特殊な目的でしか使われないので省略します。

これらの機能の詳細と意義を説明する前に、ここではプリプロセッサがどんなものかを知ってもらうために単語の置き換えの例を紹介します。リスト1を見てください。これがC言語のプログラムであるといったら何人の人が信じてくれるでしょう。先に述べたように、#defineはプリプロセッサへの指示ですし、/*と*/の間はコメントですから、リスト1のプログラムの本体は、

I think that the X68000~

の部分になります。これは単なる英文であって、プログラムではありません。しかし、このプログラムはCコンパイラでコンパイルすることができます。嘘だと思ふならコンパイルしてみましょう。X68000が2重定義だという警告は出ますが正常にコンパイルできるはず。そして、コンパイル結果を実行すると画面に、

こんにちは、世界
とプリントされます。いったい何が起きたのでしょうか。秘密は#defineにあります。

これは単語の置き換えを指示するプリプロセッサ命令なのです。この命令によって先の英文の中の単語が別の単語や記号に置き換えられ、正しいC言語のプログラムに加工されたのです（えっ、わかったの）。

試しにどのような加工が行われたのか見てみることにしましょう。XCならば/Pオプション、GCCならば-Eオプションを付けてリスト1のプログラムをコンパイルしてみてください。XCの場合は拡張子が.pのファイルに、GCCの場合は標準出力（大抵は画面上）に加工結果（プリプロセッサの出力）が出てきます。これを見ると、

```
main() { printf ("……") ; }
```

と変換されているのがわかるでしょう。これは正真正銘のC言語のプログラムですね。まあ、実際のプログラムではここまで極端な（ふざけた）置き換えは行われませんが、プリプロセッサはソースプログラムがC言語で書かれたプログラムでなくても機械的にそれを加工していることに注目してください。プリプロセッサはC言語の文法とは無関係なのです¹⁾。ただし、プリプロセッサによって加工されたあとのプログラムはC言語の文法に合致していなければなりません。

結局、Cコンパイラの前処理として使用されるプリプロセッサの役割とは、なんらかの目的（読みやすくする、二者択一すべき表現を同時に記述する、など）でC言語の文法をはずれて記述されたプログラムをCコンパイラが理解できるように加工することといえるのです。と、この程度のことを基礎知識として理解したうえで、いよいよプリプロセッサの個々の機能の説明をしましょう。ただし、今回はページ数の都合で#define、#ifdef、#includeの3命令に焦点を絞って説明します。

1) XCのバージョン2では直接オブジェクトコードを出力するが、内部的にはいったんアセンブリ言語のプログラムを作り、それをアセンブルしているだけ。

2) XCのバージョン2もプリプロセッサとコンパイラの一体型である。ただし、コンパイル速度はXCのバージョン1より2倍以上遅い。GCCで最適化付きでコンパイルするよりも遅い。

3) だから汎用のファイル変換（あるいはマクロ処理）プログラムとして利用できる。

みんな使っているマクロ機能

プリプロセッサの機能でもっともよく使用されるのが文字列の置き換え機能です。これは指定された単語を別の単語や記号に

置き換える機能⁴⁾です。単語を置き換えるためのプリプロセッサへの指示としては、

```
#define
```

を使います。この機能は先のリスト1のプログラムで体験済みですね。ただしプリプロセッサでは、リスト1のような単純な単語の置き換えだけでなく、もう少し賢い置き換え（引数を持てる）もできます。全体として#defineはマクロアセンブラのマクロと同等の機能を持っていますから、単語の置換機能はプリプロセッサでもマクロ機能と呼ばれています。

図1に#defineを使用するための形式を示します。図を見てわかるように#defineにはオブジェクト形式と関数形式の2つの形式が存在します。

オブジェクト形式のマクロは単なる単語の置き換えです。次の例を考えましょう。

```
#define TEISUU 10000
```

は置換リストが10000であるTEISUUというマクロを定義することを意味します。このときプリプロセッサが、

```
x=TEISUU ;
```

という文を見つけると、

```
x=10000 ;
```

という置換が行われます。

このようにオブジェクト形式のマクロは定数の値を指定するために使われることが多いようです⁵⁾。定数の値をマクロで定義しておけば、プログラム中で使用している定数に変更が生じた場合、そのマクロを定義している1行を変更して再コンパイルすればプログラム全体の変更を簡単に行うことができます。また、値が変更されることがないと保証されている定数に関しても意識的にマクロ化されていることがあります。たとえば、ほとんどの人がプログラムの最初で無意識に取り込むstdio.hというファイル（ファイルの取り込みに関してはあとで説明）の中ではEOF（End Of File：ファイルの終わり）とNULL（Null：何もない）という定数が、

```
#define EOF (-1)
```

```
#define NULL 0
```

というように定義されています。このマクロを使用してC言語のプログラムを、

```
if((c=getchar()) != EOF){……
```

とか、

```
while(*ptr++ != NULL){……
```

といった具合に記述すれば、その意図するところをよりいっそう明確にすることができます。

次は関数形式のマクロです。これは引数を持った単語の置き換えです。引数はC言

語での関数の定義と同様に（ ）内にカンマで区切ることで指定します。たとえば、

```
#define MAX(X,Y)((X>Y)? X : Y)
```

というマクロ定義を考えましょう。これは2つの引数X、Yを持つMAXというマクロを定義することを意味します。置換リストは、

```
((X>Y)? X : Y)
```

ですが、この中のXとYはそのままMAXの引数であるXとYに対応して置き換わります。つまり、引数の値に応じて単語（この場合は関数呼び出しのように見える）の置き換わり方が違ってきます。たとえば、

```
a=MAX(1,2);
```

```
a=MAX(a,b+1);
```

という文は、それぞれ、

```
a=((1>2)? 1 : 2);
```

```
b=((a>b+1)? a : b+1);
```

というように置き換わります。この関数型マクロは簡単な関数をプログラムの中で展開してしまうような場合に使用します。先のMAXというマクロは、

```
MAX(X,Y)
```

```
int X,Y;
```

```
{
```

```
return((X>Y)? X : Y);
```

```
}
```

というように関数として定義してもプログラムの動きとしてはほぼ同じです。しかし、マクロのほうが実際に関数の呼び出し処理がない分だけ高速です。さらに、関数として定義してしまうと引数として渡せるデータの型がintなりdoubleなりに固定されてしまいます。マクロならば単純に置換を行うだけですから引数のデータ型がなんであっても同一のマクロで処理できます。このように関数型マクロは関数にはない意味を持っているのです。

リスト2にマクロ機能の例を示しましょう。プログラムの動作はこのプログラムをコンパイルして実行すればわかると思いますので省略します。リスト2ではYESとNOをオブジェクト型マクロとして定義し、C_INPUTとSELECTを関数型マクロとして定義してありますね。それぞれがプログラムの中でどのように展開されるか（置

図1 #define（マクロ定義）の形式

- 1) オブジェクト形式
#define 識別子 [置換リスト]
- 2) 関数形式
#define 識別子 ([仮引数の並び])
[置換リスト]

[]内は省略可能

き換えられるか) 考えてみてください。

ところで、`#define`によるマクロ定義は1行に書くのが原則ですが、1行に書ききれない場合は`≡`を付けることで定義を次の行に継続することができます。リスト2のSELECTのマクロ定義がその例です。また、マクロの呼び出し(使用)は1行に書く必要はありません。もともとC言語には行の概念がありませんからそんな制限があったのでは不便でなりませんね。実際、リスト2のSELECTは呼び出し側が複数行にわたっています。これがどのように展開されているのかコンパイラの/Pオプション(GCCでは-Eオプション)で見ても楽しいでしょう。

4) ANSI Cでは置き換える対象となるのはプログラム中のコメント、文字定数、文字列以外の部分に限られる。XCではプログラム全体。

5) このようなマクロ(値に名前を付ける)はmanifest(マニフェスト)定数と呼ばれる。知ったかぶりするのに使おう。

お得な条件付きコンパイル

次は条件付きコンパイルです。これはソースプログラムの一部分をいくつかの選択肢に分けておき、条件にしたがってどれかひとつを選択するというものです。そのためのプリプロセッサ命令が`#ifdef`、`#else`、`#endif`です。これはあるマクロ(`#define`で定義するやつ)が定義されているかいないかによって実際にコンパイルすべき部分を選択します⁶⁾。このとき、マクロが定義されている場合に選択するための命令が`#ifdef`、マクロが定義されていない場合に選択するための命令が`#ifndef`です。

`#ifdef`命令の基本形式を図2に示します。基本形式1では`#ifdef`で参照するマクロが定義されている場合、`#ifdef`と`#e`

`ndif`までの間をコンパイルの対象とします(ソースプログラムに取り込む)。もしマクロが定義されていないければその部分を無視します。基本形式2ではマクロが定義されている場合は`#ifdef`と`#else`の間をコンパイルの対象とし、マクロが定義されていない場合は`#else`と`#endif`の間をコンパイルの対象とします。たとえば、

```
#ifdef __GNUC__
#define MAXSIZE 1000
#else
#define MAXSIZE 500
#endif
という記述は、__GNUC__というマクロが定義されている場合は、
#define MAXSIZE 1000
というソースプログラム(の一部)になり、
__GNUC__というマクロが定義されていない場合は、
#define MAXSIZE 500
となります。また、#ifdef命令は入れ子にして使うこともできます。
```

`#ifdef`命令はプログラムを実行するマシンやコンパイラの差異によって、C言語のソースプログラムを書き分ける必要がある場合によく使用されます。`#ifdef`命令が参照するマクロは先に説明した`#define`命令によって定義するのが普通です⁷⁾。しかし、Cコンパイラにはあらかじめ定義済みのマクロというものがあり、それを利用することもあります。たとえばXCでは、

```
Human
X68000
などのマクロが、GCCでは、
__GNUC__
mc68000
x68k
X68000
__human68k__
```

リスト2

```
1: /*      プリプロセッサ機能(その1)
2:
3:
4:      マクロ機能(文字列の置き換え)
5: */
6: #define YES      'y'
7: #define NO       'n'
8: #define C_INPUT(mess,var) printf("%s? ",mess),scanf("%c",&var)
9: #define SELECT(var,case1,mess1,case2,mess2,mess3)      ¥
10:      if(var==case1) printf(mess1);                      ¥
11:      else if(var==case2) printf(mess2);                 ¥
12:      else printf(mess3)
13:
14: main()
15: {
16:      char answer;
17:
18:      C_INPUT("中森 章は好きですか(y/n)",answer);
19:      SELECT(answer
20:              ,YES,"いやあ、照れ臭いなあ。¥n"
21:              ,NO,"カーン、悲しい……。¥n"
22:              ,"好きか嫌いかはっきりしてよ! ¥n");
23:
24: }
25:
```

などのマクロが定義済みです⁸⁾。これらの違いを利用すればコンパイラの違いに依存した部分を1本のソースファイル内に共存させることができますね。このほかに、コンパイル時にマクロの定義をすることもできます。XCの/D、GCCの-Dがそのためのオプションで、

```
/DOSK      (あるいは-DOSK)
というオプションを付けてコンパイルすれば、ソースファイルの先頭に、
```

```
#define OSK
があるのと同じ効果を持たせることができます。また、マクロ定義で置換リストを与えたいときは、
```

```
/DSIZE=100
というように=のあとで指定します。これはファイルの先頭に、
```

```
#define SIZE 100
があるのと同じです。ただこの方法で指定できるのは、いわゆるmanifest定数だけのようです。
```

それでは`#ifdef`の実例をリスト3に示しましょう。リスト3は指定したフィボナッチ数列の項を計算するプログラムです。ここではVALUEとSIMPLEという2つのマクロ名が参照されています。

VALUEは計算するフィボナッチ数列の項数を定義します。コンパイル時に/Dオプション(GCCでは-D)でVALUEに値(項数)が設定されないと、第24項を計算するようになっています。その指定をリスト3のどの部分で行っているのかはわかりますね。

SIMPLEはフィボナッチ数列の計算方法を指定します。コンパイル時にSIMPLEが/Dオプションで定義されていると、単純な再帰呼び出しでフィボナッチ数列を計算します。SIMPLEが定義されていないと、同じ再帰呼び出しですが、もう少し複雑な方法でフィボナッチ数列を計算します。

図2 `#ifdef` / `#ifndef`
(条件付きコンパイル)の形式

- 基本形式1
`#ifdef` 識別子
:
`#endif`
- 基本形式2
`#ifdef` 識別子
:
`#else`
:
`#endif`

`#ifndef`は`#ifdef`の逆条件

こちらはかなり高速です。みなさんもVA
LUEやSIMPLEを適当に定義してコンパ
イルしてみても遊んでみてくださいね。

6) 条件付きコンパイルには、このほかに、条件式
が真か偽かによってコンパイルする部分を選択す
る #if, #elif, #else, #endif もあるが、あまり使
われない。

7) 関数型のマクロが定義されているかどうかで選
択する場合は、#ifdef では引数の部分を抜かして
参照する。

```
例) #define INCI(X) (X+1)
      #ifdef INCI
      :
      #endif
```

8) マクロは名前が定義されているだけで値（置換
リスト）はない。#ifdef でのマクロ定義でも置換
リストがないものがあるが、大抵が #ifdef での参
照用である。

別のファイルを取り込む

さて、今度はファイルを取り込む機能に
ついて説明しましょう。プリプロセッサは
ソースプログラムの指定した位置に別のフ
ァイルを読み込むことができます。そのた
めの命令が #include で、使用するときの
形式は図3のようになっています⁹⁾。#in
clude 命令は、取り込むファイルの内容を
#include がある位置にそのまま書き写し
たのと同じ効果を持ちます。このため、ソ
ースプログラムの行数を短くできるという
利点があります。プログラムで必ず書くよ
うな共通の宣言などは #include で取り込
むほうがいいでしょう。

#include 命令の最も一般的な使用方法
はヘッダファイルの取り込みです。ヘッダ
ファイルとは、ライブラリで提供されてい
る関数のプロトタイプ¹⁰⁾ および戻り値、
マクロの定義をしてあるファイルです。シ
ステムであらかじめ提供されるヘッダフ
ァイルには、.h という拡張子が付いていて、
必要に応じて #include 命令で取り込める
ようになっています。たとえば、プログラ
ムの先頭でオマジナイのように書くこと
になっている

```
#include <stdio.h>
```

は、入出力のための関数のプロトタイプ、F
ILE というデータ型の定義、いくつかのマ
クロの定義を取り込むためのものです¹¹⁾。

図3 #includeの形式

- 形式1
#include <ファイル名>
- 形式2
#include "ファイル名"

リスト3

```
1: /*
2:      プリプロセッサ機能（その2）
3:
4:      条件付きコンパイル（ifdef 系）
5: */
6: #ifndef VALUE
7: #define VALUE 24
8: #endif
9:
10: fibon(n)
11: int n;
12: {
13: #ifdef SIMPLE
14:     if( n<3 ) return(1);
15:     else return( fibon(n-1)+fibon(n-2) );
16: }
17: #else
18:     return( fibl(1,n,0,1) );
19: }
20:
21: fibl(indx,max,val,va2)
22: int indx;
23: int max;
24: int val;
25: int va2;
26: {
27:     if(indx>=max) return( va2 );
28:     else return( fibl(indx+1,max,va2,val+va2) );
29: }
30: #endif
31:
32: main()
33: {
34: #ifdef SIMPLE
35:     printf("単純な再帰：フィボナッチ数列の第 %d 項は ", VALUE);
36: #else
37:     printf("複雑な再帰：フィボナッチ数列の第 %d 項は ", VALUE);
38: #endif
39:     printf("%d です。\\n", fibon(VALUE));
40: }
41:
```

リスト4

(a)

```
1: /*
2:      プリプロセッサ機能（その3）
3:
4:      ファイルの取り込み
5: */
6: #include "mydef.h"
7: #include "myfunc.c"
8: main()
9: {
10:     double x,sq0,sq1,sq2;
11:
12:     D_INPUT("浮動小数点",x);
13:
14:     sq0=sqrt(x); /* ライブラリより */
15:     sq1=fsqrt(x); /* myfunc.c より */
16:     sq2=fsqrt3(x);
17:
18:     printf("sqrt (%6.1f)= %f vs %f\\n",x, sq0, sq1);
19:     printf("sqrt3(%6.1f)= %f\\n", x, sq2);
20: }
```

(b) mydef.hの内容

```
1: #include <math.h>
2: #define YES 1
3: #define NO 0
4: #define D_INPUT(mess,var) printf("%s? ",mess),scanf("%f",&var)
5: #define F_INPUT(mess,var) printf("%s? ",mess),scanf("%f",&var)
```

(c) myfunc.cの内容

```
1: /*
2:      浮動小数点のライブラリを作ってみた。
3:
4:      注意：このプログラムを float 型のみで書くと
5:      X C ではおかしくなるよん。
6:      G C C では OK なんだな。こわい、こわい。
7: */
8: double fsqrt(x) /* ニュートン法による平方根 */
9: double x;
10: {
11:     int i;
12:     double y0,y1=0.5*(x+1.0);
13:     for(i=0;i<100;i++){
14:         y0=y1; y1=0.5*(y0+x/y0);
15:         if(y0==y1) break;
16:     }
17:     return(y0);
18: }
19:
20: double fsqrt3(x) /* ニュートン法による立方根 */
21: double x;
22: {
23:     int i;
24:     double y0,y1=(2.0*x+1.0)/3.0;
25:     for(i=0;i<100;i++){
26:         y0=y1; y1=(2.0*y0+x/(y0*y0))/3.0;
27:         if(y0==y1) break;
28:     }
29:     return(y0);
30: }
```


それでは#includeを使用する例をリスト4(a)に示しましょう。#include命令で取り込んでいるmydef.hとmyfunc.cというファイルの内容は、それぞれリスト4(b)、リスト4(c)に示してあります。これはニュートン法によって与えられた実数の平方根と立方根を求めるプログラムで、平方根についてはライブラリのsqrt関数と値を比較するようになっています。これ以上の説明は不要でしょう。ところで、リスト4(b)のmydef.hの中でmath.hを取り込んでいますが、このような多重の取り込みも可能です。ANSI規格では最低8レベルの取り込みが保証されています。

9) XCではファイル名を<>で囲むか" "で囲むかによって別の意味を持つ。つまり<>で囲んだ場合はまず標準的なディレクトリ(環境変数で決められている)でファイルを探してからカレントディレクトリを探す。一方、" "で囲んだ場合はカレントディレクトリから探す。ただし、ANSIでは、" "で囲もうが、<>で囲んだ場合と区別されていない。

10) 関数の引数(仮引数)の数とデータ型の宣言をプロトタイプ宣言という。広義には関数の戻り値もプロトタイプに含まれる。これによってCコンパイラはプログラム中で呼び出される関数の引数の数とデータ型、戻り値をチェックし、違反があると警告またはエラーを通知してくる。

11) したがって、入出力関数を使わない場合はstdio.hの取り込みは不要である。また、入出力関数を使っている場合、関数のプロトタイプ宣言をしないつもりならFILE型を使用していない限りstdio.hの取り込みは必要ない。ただ、FILE型を使わないように見えるgetcharとputcharは、FILE型を使用するマクロとしてstdio.hの中で定義してあることもあり、その場合はstdio.hを取り込まないと都合が悪い。XCではgetcharやputcharはマクロではない(マクロにすることもできるが)ため、stdio.hの取り込みが必要な場合はあまりない。

ちょっとだけANSI機能

バージョン2が発売になって、XCもよりANSI準拠になったようですから、ここでANSI規格で定められているプリプロセッサの機能を少し体験しておきましょう。ANSI規格に準拠するCコンパイラではあらかじめ__STDC__というマクロが定義されています(XCのバージョン2でも定義されています!)。このマクロを#ifdef命令で使用すればANSI規格に準拠した機能とそうでない機能の使い分けをすることができます。

ANSI規格でもっとも興味深いのは#という演算子です。これは#defineによる関数型マクロの置換リスト内で使用することができ、マクロの引数に#を付けると引数が文字列として加工されます。たとえば、

```
#define MOJI(X) (#X,X)
という定義があるとき、
MOJI(A+B)
という表現は、
("A+B",A+B)
と加工されます。すなわち、#を付けた部分は引数の値を" "で囲んだもので置き換えられます。
```

それでは、この機能がなぜ嬉しいのでしょうか。これはANSI規格で新たに規定された文字列の連結機能と関係があります。文字列の連結機能(これはプリプロセッサの機能ではありませんが)とは2つ以上の文字列を並べて書いたとき、それが1つの文字列と認識される機能です。たとえば、

```
"ABCD" "EFG"
```

は、

```
"ABCDEFGH"
```

と同じ意味を持ちます。

この機能をマクロで利用すれば文字列の一部をマクロへの引数の値にしたがって変更することができるようになります。たとえば、

```
#define SUKI(DARE). #DARE "
が好き"
```

というマクロ定義があるとき、

```
SUKI(海)
```

という記述は、

```
"海" "が好き"
```

と置き換えられ、これは

```
"海が好き"
```

と同じことになります。このような文字列の加工はprintf関数の書式指定などで威力を発揮するでしょう。

リスト5に#演算子を使用したプログラムの例を示します。これはXCのバージョン

1では正常にコンパイルできないので注意してください。リスト5で注目してほしいのはDISPLAYというマクロでの#の使われ方です。これは式の値をプリントするためのマクロですが、引数として与えた式からprintf関数に渡す書式と値をそのまま作り出しています。たとえば、

```
DISPLAY(a+b,%d);
```

は、

```
printf("a+b" "=" "%d" "n", a+b);
```

と置き換えられて、結局これは、

```
printf("a+b=%dn", a+b);
```

と同じになります。つまり、引数として与えた式にしたがって、どういう式の値かを表示するようにできるのです。ANSI規格以前ではこのような芸当は不可能でしょう。

*

C言語のプリプロセッサ機能の主なもの概観してきました。C言語の学習の中でプリプロセッサの機能は忘れ去られる傾向にあります。自分でプログラムを書く場合にはプリプロセッサをほとんど知らなくてもなんとかなるでしょう。しかし、他人の書いたプログラムを読みこなすためにはプリプロセッサの知識は必須です。また、自分のプログラムを読みやすく、効率よく書くためにもプリプロセッサは大いに役立つことでしょう。これを機会に読者の皆さんも自分なりのプログラミングスタイルを考えてみるのもいいかもしれませんね。

【参考文献】

- 1) 林晴比古, Cプリプロセッサ・パワー, 日本ソフトバンク, 1988年.
- 2) 平林雅英, ANSI C言語事典, 技術評論社, 1989年.
- 3) マーク・ウィリアムズ(編), ANSI C言語大事典, パーソナルメディア, 1990年.

リスト5

```
1: /*
2:     プリプロセッサ機能 (その4)
3:
4:     ちょっとANSIな機能
5:     XCのバージョン1ではコンパイルできないのねんねん。
6: */
7: #define G_INPUT(mess,type,var) printf(mess "%2 ",scanf("%type", &var))
8: #define DISPLAY(expr,type) printf(expr "=" "%type %n", expr)
9:
10: #ifdef __STDC__
11: char date[]=__DATE__ "/" __TIME__; /* 文字列の連結が起きる */
12: char fname[]=__FILE__;
13: #else
14: char date[]="コンパイル日時是不明です.";
15: char fname[]="ファイル名は不明です.";
16: #endif
17:
18: main()
19: {
20:     int a,b;
21:     printf("ソースファイル:%s\n", fname);
22:     printf("コンパイル日時:%s\n", date);
23:
24:     G_INPUT(a, %d,a);
25:     G_INPUT(b, %d,b);
26:
27:     DISPLAY(a+b, %d);
28:     DISPLAY(a-b, %d);
29:     DISPLAY(a*b, %d);
30: }
```


基礎知識からプログラミングへ

Cライブラリ利用の手引き

Tan Akihiko 丹 明彦

C言語がその威力を発揮できるのは1にも2にもライブラリのお陰である。特にXCの魅力はその豊富なライブラリにあるといてもいいくらいだ。まずはCコンパイラとライブラリの熱い関係を理解して、プログラミングに挑戦してみよう。

ライブラリ。英語で書くとlibrary。図書館とか蔵書とかいう意味だ。Cプログラミングにおけるライブラリも、ほぼそういう意味だ。

XCのライブラリは豊富である。本当に豊富である。IOCSやDOSからBASICまで、X68000のあらゆるサービスがCから利用できる。ついでにいうと、このライブラリはGCC (GNU C) から利用できる。ほんの数行プログラムを書いただけで、高度な処理が可能になる。

C言語はほかの高級言語に比べて開発用の言語としてはかなり優位な立場にある。その理由のひとつはライブラリが充実しているということにあると思う。

ライブラリはいろいろな意味で、通常の高級言語にある組み込み関数とは趣を異にしている。

ライブラリとはなんぞや

さて、Cを使った人がまず確実に引っ掛かると思われる（自分を基準にしてはいけないのだが）箇所について指摘しておきたい。使ううえでは支障にならないことだが、とても大切なことである。そしてこれが、通常の（手続き型）高級言語とCとの間にある差を象徴的に表していると僕は見る。ことはCの設計思想にまで及ぶ問題なのだ。いきなり質問から始めよう。

printf()はCの命令である。○か×か。

答えは「×」である。これは冷静に考えてみると、とんでもないこととは思わないだろうか。画面に文字や数字をプリントするというプログラミング言語においてはとても基本的な機能すらも、Cの言語仕様には定義されていないのだ（例のK&Rでは、printf()を標準的な関数として、その仕様を書いてあるが、それも文法の仕様とは別のものである）。このへんにCの本質のひとつをかいま見ることができる。

Cコンパイラそのものでできることは、実はアセンブラとたいして変わらない。ア

センブラに毛の生えた程度の演算や、簡単な制御構造。これがC言語のほとんどすべてである。ただ、配列や構造体を使えるようにしていることで、複雑なデータ構造を処理できる。このおかげで、アセンブラよりはるかに変数などの管理が楽になっている。乱暴だが、Cは構造化されているアセンブラの代替品ともいえるのである。

ではprintf()とはなんなのか。ここではサブルーチンのひとつという答えを与えておこう。文字のプリントなどというものを仕様の中にきっちりとうたっているアセンブラなどあるはずもないが、それでも文字のプリントは現実に可能な処理である。適当なパラメータで呼び出せば要求どおりの文字を出してくれる、そんなサブルーチンを用意しておいて呼び出せばいい。Cでは、これをサブルーチンと呼ばずに関数と呼んでいる。

そうした関数は、Cコンパイラを入手するともれなくついてくることになっている。それがライブラリである。ユーザーがプログラム中にprintf()という関数を使用したら、コンパイラはライブラリの中からprintf()関数を引っ張り出し、ユーザーの書いたプログラムとくっつける。そうすることで、printf()があたかもCの命令のひとつであるかのように振る舞うことができるのである。

ところで、関数はユーザーが定義することもできる。main()をはじめとして、Cのプログラムは関数の集まりである。そして、Cにおいては、ライブラリに入っている関数もユーザーが書いた関数もほぼ同じ扱いを受ける。自分が書いた関数をライブラリに追加することもできる。これはほかの高級言語風にいえば、命令をユーザーが増やせるというのと同じことなのである。

とはいえ、そんな基本的な関数からいちいち作っていくのは骨の折れる仕事である。そこにライブラリの存在意義がある。Cの処理系には、必ずライブラリが付属していて、printf()などのようなごくごく基本

的な関数は、自分で作る必要はない。最初にもいったとおり、XCの場合は、およそX68000を使う上で必要になりそうな処理はすべてライブラリで準備してある。ライブラリがないCなんて、単なるダルマさんなのである。

ライブラリとヘッダと関数とマクロ

ライブラリ理解への道に、またひとつハードルが見えてきた。それはライブラリとインクルードファイルとの関係。インクルードとは、他のファイルをソースリストに取り込むこと。このへんもアセンブラの影響を強く受けているようではある。インクルード命令を上手に使うと、以前からあるソフトウェア資産を手軽に生かすことができる。それではまたも質問形式でいってみよう。

関数printf()を使うには、ヘッダファイル“stdio.h”をインクルードすればよい。これは正しいか。

答えは、「嘘ではないが正確でもない」だ（なんじゃそりや）。関数printf()は、stdio.hをインクルードしなくても使えるし、またインクルードしただけで使えるとは限らない。printf()を利用するためには、もっと別の方面から攻めなくてはならないのだ（というほど大袈裟でもない）。

種明かしをしよう。すでにガイドマップで解説されているように、ヘッダファイルでは、単に関数の外部参照宣言しかしていない。stdio.hにはプログラムらしいプログラムは書かれていない。ヘッダはあくまでもヘッダ、そこに関数の本体は書かないようになっている。関数の本体はライブラリに入っている。だから、

```
#include <stdio.h>
```

とソースリストに書くのは、この場合、

```
extern int printf();
```

と書くのとはほとんど変わらない。

stdio.hをインクルードした時点では、コンパイラは「あ、このプログラムはprintf

()をどこから呼び出してきて使うんだ」と認識するだけ。コンパイルが進行し、リンクフェイズに達して初めてprintf()の本体を探しにかかるのである。このときリンクは、プログラムがかつてstdio.hをインクルードしていたなどということは知らない。わかるのは、受け取ったプログラムのなかでprintf()が外部参照になっているということだけ。だから、ライブラリからprintf()を探し出して、メインプログラムと結合しなくてはならない。このときもしもライブラリからprintf()を見つげ出すことができなかったり、ライブラリそのものがなかったりしたら、当然エラーとなって、コンパイルは中断される。

このへんも、C言語を使っているアセンブラの香りを強烈に感じる部分である。

なお、先ほど「関数printf()はstdio.hをインクルードしなくても使える」といったが、これはprintf()がごく素直なタイプの関数だからである。Cの関数宣言では、整数(int)型のものに限り省略してよいことになっている。しかし、無用のバグの発生を避けるため、また、他の人がこのプログラムを見たとき困らないように、ヘッダファイルをインクルードする命令は面倒でも入れておくようにしよう。

XCのための開発環境

Cプログラミングにおいて、これほど大切なライブラリであるが、いったいどこにあって、どうやって利用するのだろうか。といっても気にする必要なし。Cの処理系をインストールするときに環境設定してしまえば、あとは忘れてしまってもかまわない。必要に応じて思い出すなり参考書を見るなりすればよい。

Cコンパイラを買えばシステムディスクがついてくる(当たり前)。そのディスクのディレクトリ構成を調べたりautoexec.batをよく読んだりすれば、設定のしかたはすぐにわかる。

システムディスクがAドライブだとしよう。ハードディスクでもフロッピーディスクでもかまわない。そこにライブラリやインクルードファイルを格納しておく専用のディレクトリを作る。その名もストレートにA:¥LIBおよびA:¥INCLUDE。

次に、Cコンパイラがライブラリやインクルードファイルを探しに行くときに参照する環境変数がある。その名もこれまたストレートでlibおよびinclude。これらには、ライブラリやインクルードファイルのある

ディレクトリを指定する。値をセットするにはautoexec.batを使うのが一般的である。

```
autoexec.batに、
set lib=A:¥LIB
set include=A:¥INCLUDE
```

の2行を加えておけば、以後コンパイラは自動的にライブラリやインクルードファイルのあるディレクトリを探しに行く。つまり、システムを再構築するまでは忘れていてもいいようにかまわないというわけ。

ライブラリの選択

しつこいようだがXCのライブラリは豊富である。それだけにマニュアルも分厚い。コンパイラがバージョン2になって、ライブラリマニュアルが2冊に分かれた。それでも相当な厚さである。それくらいライブラリは豊富に取り揃えてある。だからといって怖がる必要はまったくない。まさか、これを全部暗記しないとCをマスターできないのでは？ と考える方はいないだろう。何度もいうように、ライブラリはCの文法そのものではない。あくまで、とても便利なサブルーチン群なのだから、必要なものだけつまみ食いしていくのが正しい。

多すぎて使いきれないライブラリ関数のなかでも、最もよく使われていると思われるものは、なんといってもstdio.h(で宣言される関数)であろう。スタンダードI/O、標準入出力の略だ。これとまったく関わりのないCのプログラムを探すほうが難しいくらいだ。とりわけprintf()関数およびその類似関数と縁もゆかりもないプログラムは、めったにお目に掛かれるものではない。というわけで、stdio.h(で宣言される関数)の掌握は必須課題である。

あと、ちょっと細かい話になるし、たいして重要でもないが、コンパイルをつつがなく終わらせるために必要なことが少しある。

XCライブラリの関数には、レベルがつけられている。関数に格がついているのだ。レベルは0から3まである。原則的には、
レベル0: IOCS
レベル1: DOS
レベル2: C
レベル3: BASIC

となっている。レベルの数字は低水準の関数では低く、高水準になると上がるようである(ここでの低水準とは必ずしも程度の低いことを意味しない。むしろ使う側に高度なものを要求する。一般に低水準関数

のほうが使うのが難しく、それだけに高度な処理もこなせる)。これらのレベルの関数はそれぞれ、

```
IOCSLIB.A
DOSLIB.A
CLIB.A
BASLIB.A
```

というファイルに収められている。なお、XC ver.2では少々違って、ファイル名は~.Aではなく~.L、ファイル構造も異なるので互換性はない。

Cの標準関数はレベル2となっている。他の処理系への移植を考えるなら、レベル2の関数だけを利用すべきである。ハードに密着していたり、画面や音源まわりを直接ドライブするような危険なものでないならば、たいいてい処理はレベル2で十分こなせる。例のstdio.hで宣言する関数群もこのレベル2の関数である。

XCでは、CLIB.Aに収められている標準関数以外の関数を使うときには、コンパイル時に/Wや/Yといったスイッチを指定する必要がある。これらのスイッチを使わないとコンパイルできないプログラムは、ほかの処理系へ移植するのが難しいと思って差し支えない。

関数利用の注意点

さて、ひとつの関数を利用するときに、調べておいたほうがいいことはいくつかある。

・その関数の機能が本当に自分の求めるものか

これは当たり前。関数の機能がよくわからないときは、とりあえずコンパイルしてみてもいい。うまくいかなかったら別の関数にしてみる、という行き当たりばったり方式でもいい。ライブラリ関数のなかには、似たような機能を持ったものも多い。関数のレベルが違っていたりするので、安全そうなものを選ぼう。XCでコンパイルする場合、IOCS、DOS、それにBASIC関係のライブラリを利用するときにはコンパイル時にスイッチ/W(BASIC)、/Y(IOCS、DOS)が必要である。

・宣言はどのインクルードファイルで行われているか

#include命令は書く必要のない場合も多いが、デバッグや移植のことを考えると、なるべく書くようにしたほうがいいのはいうまでもない。また、インクルードファイルでは、定数や構造体の宣言をしていることも多い。たとえばよく使うstdio.hにしても、ファイル構造体FILEやエンドオブ

ファイルを示す定数EOFを宣言しており、インクルードしないままコンパイルすると、「そんなもの宣言してないぞ」と怒られる。というわけで#include命令はまめに書くようにしよう。

・関数の戻り値はなにか、引数の型はなにか、数はいくつか

関数呼び出しでは、このへんからくるトラブルがとにかく多い。Cコンパイラは引数のチェックをほとんどしない。関数プロトタイプ宣言してあれば、多少はするとはいうものの、とても「暴走しない」というレベルまで面倒みてはくれない。バリエーションやアドレスエラーが出たときは、まずこのへんを疑うのがよいだろう。とにかく型チェックは基本である。

これだけのことをチェックしたら、プログラムリストに関数呼び出しの式を書くのだが、これがまた面倒なことに、ライブラリマニュアルに書いてある書式をそのまま書き移してもうまく動かない。ライブラリマニュアルのページ構成は図1のようになっているのだが、この読み方がちょっとわかりにくいかもしれない。例はprintf()とput()。XCバージョン2のライブラリマニュアルから引用した。

実は、マニュアルに書いてあるのは、関数本体、呼び出される側の書き方である。呼び出すほうでは違う書き方を要する。つまりマニュアルの記述を多少読み換える必要があるのである。たとえば、

```
printf("ただいま処理中です¥n");
printf("(x,y)=(%d,%d)¥n", x, y);
put( x, y, x1, y1, pat, 16*16*2);
```

のように書く。これを図1と比べると、マニュアルの読み換え方が多少なりともわかりただけだと思う。一応まとめておくと、以下になる。

・関数の型であるintやvoidは書かない。

これが必要なのは関数本体（とインクルードファイルでの宣言部）だけ。

・変数名は、ユーザープログラムでの名前を使う。マニュアルに載っている名前を使う必要はない。

これは、ローカル変数のなんたるかを存じの方には無意味なアドバイス。

・[]で括られた部分の引数は、いくつ書いてもいい。

もちろん省略も可能。

・型が“void*”となっている引数には、どんな型のポインタを使ってもいい。

たとえばput()関数の場合は、bufがその変数へのポインタであれば何型でもかわらない。unsigned charやunsigned shortのことが多い。

ライブラリマニュアルはCをある程度使い慣れた人が読めるように作ってあるので、初めのうちは少々戸惑うこともある。

もっとも参考になるのは、実はページの下サンプルプログラムである。結局、習うより慣れろというのが一番いいのかもしれない。ああ陳腐な結論。

楽して実を取れ

要領のいいプログラマはライブラリを上手に使う。一見して複雑そうな処理も驚くほど短い行数のプログラムで実現できる。

優れたタッチタイピストであれば、一瞬のうちにソースファイルを書き上げることも可能である。

ライブラリの欠点は、まずなんといっても最高速でないこと。いくらCが危険性の高い暴走できる高級言語だといっても、ライブラリが平気でそんなことをやっていては話にならないわけで、ひととおりの安全設計とある程度の汎用性は必要なわけだ。そうすると、速度の落ちるのは必定。どうしてもスピードの欲しい場合には、専用のルーチンを自分で書くことになるのだが、それまではおとなしくライブラリを利用しておけばよい。ま、アフターバーナーならともかく、レイトレーシングで画面にドットを打つのにわざわざスーパーバイザモードに入って描画速度を稼ぐのも間抜けな話ではある。

図1 ライブラリマニュアルの例

printf	レベル2
書式	#include <stdio.h>
	int printf(format_string[, arg...]);
	char *format_string;
機能	format_stringで指定した…… (以下省略)
put	レベル3
書式	#include <graph.h>
	void put(x1, y1, x2, y2, buf, size);
	int x1;
	int y1;
	int x2;
	int y2;
	void *buf;
	int size;
機能	bufが示す配列に格納された…… (以下省略)

応用編

ライブラリを使って外部コマンドを作る

X68000ユーザーのほとんどがふだん使っている環境といえば、コマンドシェル(COMMAND, X)であろう。少なくともCプログラマはそのはずだ。SX-WINDOWの環境、特にCの開発環境が整うのはもう少しと先の話。Cコンパイラも、コマンドシェルでの使用を前提に考えている。

で、コマンドシェルの環境を充実させようというわけだ。特に力を入れたのは操作性の向上だ。そのためのアプローチはいくつかある。たとえば、速度の向上 (I/OCS, Xなどを使う)、ヒストリドライバの

活用、バッチファイルの作成などである。

そして新しいコマンドの作成というのもある。これが本題。ライブラリを使いまくれば、簡単に新しい処理を行うコマンドをいくつでも作ることができる。極端な場合、ひとつの関数につきひとつの外部コマンドを作ってしまう。パラメータはコマンドラインから取り込む。バッチファイルから次々と呼び出せば、インタプリタに近いことも可能である。

というわけで今月の色物。自作の外部コマンドとバッチファイルで、インタプリタもどきを作る。

プログラムには、有名な「エラトステネスのふるい」を選んだ。素数を求めるアルゴリズムの代表的なものである。今回作った外部コマンドは以下のとおり。それぞれリスト1～5となっている。

●numbers.x

指定された区間の自然数を数列にして標準出力に出力する。たとえば「numbers 15」とした場合、「1 2 3 4 5」と画面に出力される。

●null.x

標準入力から入力した数列が空かどうか調べる。もしも空なら、環境変数errorlevelに1をセットする。

●head.x

標準入力から入力した数列の先頭の数を取って標準出力に出力する。

●tail.x

標準入力から入力した数列の先頭の数を除く数列を標準出力に出力する。

●div.x

標準入力から入力した数列の1番目の数が2番目の数で割り切れるかどうかチェックする。割り切れない場合は環境変数errorlevelに1をセットする。

リダイレクションの魔術師

さて、前記のコマンド解説にはやたらと「標準入力」「標準出力」という言葉が出てくる。これはもういかにもstdio.hの出番だな。解説しよう。

通常のコマンドラインで指定したファイル名を受け取って、それに対応するファイルをfopen()で開き、fscanf()やfprintf()でアクセスする。そうしてもよかったのだが、いかんせんプログラムが長くなる。それを嫌って標準入出力を使ってみた。標準入出力とは、通常はコンソールのこと。人間が計算機を使う場合の多くは、キーボードから手で入力し、出力が画面に出てくるのを目で見ることになる。それが標準ということの意味である。

標準入出力の場合、ファイルのオープン/クローズは不要。よってそれにまつわるエラー処理も不要になり、行数が短くてできる。scanf()やprintf()などがあるので入出力したい場合にも困らない。また、標準入出力を使うようにしたおかげで、ファイルの受け渡しにリダイレクションが使えるようになった。リダイレクションとは、本来ならキーボードである標準入力の入力元をファイルに指定したり、本来なら画面である標準出力の出力先をファイルに指定したりすることである。パイプ機能と組み合わせればフィルタとして使えるようになる。

リスト1 numbers.c

```
1: /* numbers.c */
2: #include <stdio.h>
3: int main( argc, argv )
4: {
5:     char *argv[];
6:     {
7:         int a, b, i;
8:         sscanf( argv[1], "%d", &a );
9:         sscanf( argv[2], "%d", &b );
10:        for ( i=a; i<=b; i++ ) printf( "%d ", i );
11:        return ( 0 );
12: }
```

リスト2 null.c

```
1: /* null.c */
2: #include <stdio.h>
3: int main()
4: {
5:     int i;
6:     return ( scanf( "%d", &i )!=EOF );
7: }
```

リスト3 head.c

```
1: /* head.c */
2: #include <stdio.h>
3: int main()
4: {
5:     int i;
6:     scanf( "%d", &i );
7:     printf( "%d ", i );
8:     return ( 0 );
9: }
```

ただし、フィルタとして働くプログラムをリダイレクション入力（もしくはパイプ入力）指定なしで実行すると、カーソルが点滅したまま実行が止まってしまう（ように見える）。これはもちろん、標準入力、つまりキーボードからの入力を待っているからである。リダイレクションからの入力がない場合、標準入力はキーボードからとなっている。で、この状態を抜けてコマンドラインに戻りたい場合はCTRL+Z（CTRLキーとZキーを同時に押す）がよろしい。これはエンドオブファイルを表すコントロールコードで、標準入力にこれが入るとフィルタのプログラムは終了する。

また、リダイレクション出力指定のない場合、標準出力は画面であるから、処理した結果は画面に出てくることになる。

コンパイルと実行

さて、今回のプログラムは、標準入出力（実際はリダイレクトするファイル）を線形リストとみなしたリスト処理プログラムになっている。さっそくプログラムを打ち込んでコンパイルしよう。これ以上短くてできないところまで短くしたので、打ち込みにたいして時間はかからないと思う。特殊な関数もいっさい使っていないので、単に

CC ~.C

とするだけでいいはずだ。

では、デバッグも兼ねて、動作テストを

リスト4 tail.c

```
1: /* tail.c */
2: #include <stdio.h>
3: int main()
4: {
5:     int i;
6:     scanf( "%d", &i );
7:     while ( scanf( "%d", &i )!=EOF ) {
8:         printf( "%d ", i );
9:     }
10:    return ( 0 );
11: }
```

リスト5 div.c

```
1: /* div.c */
2: #include <stdio.h>
3: int main()
4: {
5:     int a, b;
6:     scanf( "%d %d", &a, &b );
7:     return ( (a%b)!=0 );
8: }
```

図2 動作テスト

```
1: B>numbers 3 10
2: 3 4 5 6 7 8 9 10
3: B>numbers 3 10 | head
4: 3
5: B>numbers 3 10 | tail
6: 4 5 6 7 8 9 10
7: B>numbers 3 10 | tail | null | if errorlevel | echo "nullです"
8: B>numbers 3 3 | tail | null | if errorlevel | echo "nullです"
9: "nullです"
10: B>div < con | if errorlevel | echo "割り切れません"
11: 4 2 ^Z
12: B>div < con | if errorlevel | echo "割り切れません"
13: 5 3 ^Z
14: "割り切れません"
```

※ ^ZはコントロールZのこと。CTRL+Zキーを押して入力する。エンドオブファイルを表す。

してみよう。図2のアンダーライン部をコマンドラインから入力して、出力が正しいことを確かめてもらいたい。出力のうち改行はこの図のとおりにはいらないが、そのへんはよきにはからっていただきたい。それでは解説の必要がありそうなところをちょこちょこつまんでみる。

* 第3行：見慣れない記号“|”がある。Cでは、これは「または」だったが、コマンドシェルでは全然違っていて、いわゆるパイプである。左のコマンドの標準出力を横取りして、右のコマンドに渡す働きをする。この場合は、numbers.xの出力“345678910”をhead.xに渡している。head.xはその先頭の数字を取るのだから“3”と表示されるのである（第4行）。

第7行：パイプを2段につないでいる。numbers.xの出力“345678910”をtail.xに通すと“45678910”になる。これはもちろん空リストではないので、環境変数errorlevelは0のままである。なお、記号“|”は、BASICでいうマルチステートメントで、リダイレクションとは関係ない。

第8行：numbers.xに与える2つの引数が同じである。したがって出力は“3”だけ。それをtail.xに通すと、先頭以外の要素がないので、当然空リストが出力される。するとnull.xのチェックに引っ掛かってerrorlevelに1が立ち、第9行のメッセージとなる。

第10行：div.xの入力に“con”というファイルをリダイレクトしている。これはコ

ンソールの略で、キーボードから数字を入力させる。それが第11行の“4 2”である。4は2で割り切れるので、errorlevelは0。

第13行：今度は5が3で割り切れないのでerrorlevelが1になり、割り切れないというメッセージをもらう（第14行）。

*

コーディング上の細かい話をひとつしておこう。null.cやtail.cで、エンドオブファイルをfeof()で見ずにscanf()の戻り値がEOF（実際の値は-1。stdio.hで定義してある定数である）かどうかで見ていますが、これには理由がある。ファイルの終わりにスペースや改行などの空白文字が残っているときに、feof()はエンドオブファイルとは見えないが、scanf()は（もうファイルに数字が残っていないので）エンドオブファイルと見てくれる。数列なのだから、空白文字は無視してほしいというわけ。

*

リダイ렉션には、上で使ったもののほかにも追加リダイ렉션というものがある。

```
type _a >> _b
とすれば、ファイル_bの後ろにファイル_aの内容がつながる。BASIC風にいえば、
_b = _b+_a
のようなものだ。
```

もうひとつ、これはリダイ렉션ではないが、copyコマンドにはファイルを結合する機能もある。上と同じことは、

```
copy _b+_a _tmp||copy _tmp _b
とすればできる（同じといっても、画面に出てくるメッセージは少々違う）。この2つは覚えておくとも便利かもしれない。
```

COMMAND.Xはプログラミング言語？

さて、デバッグもすんだリスト処理コマンドを利用して、そろそろエラトステネス

のふるいを作ってみよう。エラトステネスのふるいの考え方を図3に示す。

素数とは1とそれ自身以外に約数のない数のことである（1は素数ではない）。最初に2以降の数列を作っておく。数列の先頭の数は常に素数である。先頭の数を素数リストに追加する。数列の先頭を取った残りから、先頭の数の倍数を取り除く。すると、残った数列の先頭もまた素数になる。これを繰り返す。

というわけで、リスト6のprime.batを打ち込んで実行していただきたい。先ほどのリスト処理関数群と同じディレクトリに置いて、できればRAMディスクで動かしたほうがいい。気が遠くなるほど遅いし、ディスクアクセスも頻繁だからだ。実行ファイル名はいうまでもなくprime（primeとは素数のことである）。遅いので、最初は、prime 10のように小さい数から実行してみることを

望ましいコマンドの書き方

Cはコマンドラインから引数を受け取れるようになっていて、これがもう便利なのである。さすがにUNIX由来だけのことはある。ほかのプログラミング言語でコマンドラインから引数を受け取る（これを簡単にもらえるものを僕はあまり知らない（アセンブラは別だが）。たいしてはいきなりINPUT文かなにかで、「ファイル名を入力してください」ということになっている。以下、よりよい操作環境のためのコマンドに身をつけてほしい条件を、順不同に列挙しよう。

*

対話形式プログラムの最大の欠点は、バッチファイルの中に書けないことであろう。対話形式プログラムが走り出すと、そこでバッチファイルの処理が中断し、ユーザーのキー入力をつまでも待っているからである。先ほどのINPUT文はその典型。有名なところとしては、少し前までのMS-DOS。起動するたびに今日の日付と時刻を尋ねてきていた。要するにAUTOEXEC.BATにDATEやTIMEと書いてあっただけのことなのだが、いまどき内蔵時計のバッテリーバックアップは当たり前なのだから、これはなかなか凶悪な仕打ちである。リターンキーでスキップすればいいということ知らなかったら、起動するたびに日付と時刻を調べて入力しなすなどという間抜けなことにもなりかねない。対話形式にする必要が特にないプログラムは、対話形式にしないほうがずっと使える。

逆に、ハードディスクのフォーマットのように、滅多に使うものではなく、また自動実行されると困るようなものは、むしろ対話形式のものがよい。誤操作でデータがごっそり消えてしまうのを防ぐことができるからだ。また、それ単独で「シェル」と呼べるもの、ひとつの環境を作っているものは、対話形式でもいい。かなり不満ではあるが、MENU.Xも用途次第ではまあ許せる。対話形式を避けるべきなのは、あくまでコマンドシェルから呼び出すことを前提にしているツール類の話である。

*

複数の引数を取りうる時、その順序は任意にできるのが望ましい。特にスイッチの処理は、書く順序が限定されていると使いにくいことこの上ない。ひとつのスイッチに複数のオプションを記述する（たとえば“-xvf”のように）ことができるとうれしい。

*

ファイル名の受け取り方にも柔軟性がほしい。処理するファイルの拡張子がわかっているような場合、拡張子を省略したり、つけたり、どちらもできるとうれしい。

*

引数が必要なにつけなかったとき、またはスイッチなどの使い方を間違えたときには、コマンドの使い方を簡単にでもいいからプリントアウトするのが親切。MS-DOSのコマンドは、「××はできません」「文法が違います」といったエラーメッセージが出てきておしまいということが多かった。それならどうすればいいのか。アドベンチャーゲームではないのだ。

これがUNIXだと少々違う。UNIXは確かに無愛想なOSだが、コマンドの実行に失敗したときは使い方を教えてくれる。詳しいオンラインマニュアルもあるが、使わなくてもなんとかなる。ついでにいておく、Human68kのコマンドは使い方を教えてくれるものが多い。PC-9801をX68000と同じ感覚で使っていていちいち引っ掛かるのはこのへんにも原因があるのかもしれない。

*

エラトラップにも細心の注意を払いたい。暴走するのは最低だ。特に、引数の個数をしっかりチェックしていないコマンドは、使い方を間違えるとたちまちバスエラーやアドレスエラーの洪水だ。また、引数1にはこれを、引数2にはあれを、……というプログラムの決めた仕様をそのままユーザーに押し付けてもいいが、そのときはマニュアルを充実させること。それよりもプログラムの都合を押し付けけない心掛けがむしろ大切。

*

フィルタになりうるもの、特にフィルタとして使うとおいしいものは、標準入力と標準出力をうまく使ってフィルタにすること。フィルタとしてでなく使いたい場合も考えられるものは、「フィルタになるスイッチ」を設けてもいいし、引数の指定のしかたで動作を切り替えてもいい。このタイプのプログラムとしてはMORE.Xが代表的。

DIR | MORE

とすればフィルタとして、

MORE test.c

とすればページ単位のTYPEとして働く。

*

グラフィックRAMを使うもの、マウスを使うもの、その他ハードウェアに密接するものは、へたをするとユーザーを混乱に陥れるので要注意。典型的な例を挙げる。

グラフィックRAMをRAMディスクとして使っている場合には、グラフィックを勝手に使ってRAMディスクを破壊しないようにする必要があるが、かといって、RAMディスクをテンポラリのドライブにしかしていないなら、使えないのも困りもの。とくに、グラフィックを使うとわかっているプログラム（画像ロードなど）なら、起動した時点でグラフィックを使うというユーザーの意志はあるわけで、それでも「グラフィックRAMは使えません」の一点張りではあまりにお粗末。メモリを拡張していたらどうということはないが、512Kバイトはとても魅力的な記憶容量、最大限に活用したいものだ。コンパイルのときはテンポラリファイルの置き場に、実行のときはグラフィックRAM本来の機能になってほしい。

だから、強制的にグラフィックRAMを使う（内容を破壊してもいい）スイッチ、またはグラフィックRAMの使用状態をチェックするスイッチを用意すると親切だな。といったつ、今回のサンプルではそれをさぼっていたりする。

以上、自戒をこめて、外部コマンド作成の心得をおしまいにす。

勧める。ちなみに、100まで求めたら、RAMディスクで6分ほどかった(!)。

リストの解説は、読者の皆さんにお任せしよう。Human68kのマニュアルを読めばすべて書いてあることだ。バッチファイルなのに生意気にもインデントなどして、高級言語のふりをしている。

まあ、この程度のプログラムはすべてCで書いたほうがはるかに簡単だし、高速である。この「コマンドラインインタプリタ」は、実際呆れるほど遅い。もう少し速かったら、ドライトンベンチマークやら、果てはレイトレーシング(!)までするつもりだったが、どうも無理のようだ。

しかし、あえてCで書くのは最低限の基本コマンドだけに抑え、できる限りバッチファイル用の命令でプログラムを書いたのだ。試みとしてはけっこう面白い。まず、変数をファイルで持ったのがいい。今回はリストがその変数にあたる。いきなり可変長のデータ構造をサポートしているのである。それにハードディスクで実行すれば、停電しても変数の値が消えない(?)。

バッチファイルならばトレースも簡単。オールCで書いた場合、ソースコードデバッガなしでデバッグをしようと思ったら、ソースにprintf()などを挿入してコンパイルしなおさなくてはならない。デバッグが終わったときは、さっき入れたprintf()を削って再コンパイルだ。極めて面倒。しかしバッチファイルなら、typeなどをちょっと挟むだけですむ。プログラムによってはそれをdumpにしてもいいし、findだろうがdbだろうが思いのまま。要するにアルゴリズムにあわせて好きなデバッグ用ツールを使い放題にできるのだ。デバッグが終わっても、その行を削除するだけ。要するに、基本コマンドさえ充実しておけば(重要)、あとはインタプリタとはほとんど同じ感覚で使えるのである。

BASICライブラリもお得

最後におまけとして、BASICライブラリを用いたコマンドサンプルをつけておく。ここで述べてきたことも多少は生かしている、参考にしていただきたい。

簡単な画像処理コマンドである。名前はACCENT.X。65536色モードで画像をロードしておき、ACCENT.Xを実行すると、色が強調される。何度もエフェクトをかけると、しまいには原色(8色!)になって楽しい。約2年前(1988年9月号)の画像処理プログラムのうち、色強調処理を取

り出して高速化した(といっても、XCでコンパイルした場合とGCCでコンパイルした場合とでは、速度に雲泥の差がある)。

コンパイルは、ACCENT.CがBASICライブラリを使っている、次のように/Wスイッチをつける。

図3 エラトステネスのふるい

```
1:      2 3 4 5 6 7 8 9 10 11  始めの状態。
2: 2    ← 3 5 7 9 10 11  先頭の2を素数リストに追加し、残りの数値から2の倍数を取り除く。
3: 2 3   ← 5 7 9 10 11  以下繰り返す。
4: 2 3 5   ← 7 9 10 11
5: 2 3 5 7   ← 9 10 11
6: 2 3 5 7 11 ←      結果
```

リスト6 prime.bat

```
1: ECHO OFF
2: REM エラトステネスのふるい
3: IF NOT "%1" == "" GOTO START
4: ECHO 使い方: PRIME [n]
5: ECHO n までの素数を求めます。
6: GOTO END
7: :START
8: numbers 2 %1 > _n
9: ECHO 素数: > prime.lst
10: :LOOP1
11: null < _n || IF ERRORLEVEL 1 GOTO RESULT
12: head < _n > _h
13: tail < _n > _t
14: rem TYPE _h
15: COPY prime.lst+_h _tmp > NUL || COPY _tmp prime.lst > NUL
16: COPY NUL _n > NUL
17: :LOOP2
18: null < _t || IF ERRORLEVEL 1 GOTO LOOP1
19: head < _t > _a
20: tail < _t > _t
21: COPY _a+_h _b > NUL
22: div < _b || IF ERRORLEVEL 1 COPY _n+_a _tmp > NUL || COPY _tmp _n > NUL
23: GOTO LOOP2
24: :RESULT
25: DEL /Y _* > NUL
26: TYPE prime.lst
27: :END
```

リスト7 あると便利な色処理マクロcolor.h

```
1: #define RED(C) ((C)>>6) & 31
2: #define GREEN(C) ((C)>>11) & 31
3: #define BLUE(C) ((C)>>16) & 31
4: #define RGB(R,G,B) ((R)<<6 | (G)<<11 | (B)<<16)
```

リスト8 画像の色を強調するaccent.c

```
1: #include <ioclib.h> /* 関数 CRTMOD() のためのインクルードファイル */
2: #include <stdio.h> /* 関数 fprintf() */
3: #include <stdlib.h> /* 関数 get(), put(), put() */
4: #include "color.h" /* マクロ RED(), GREEN(), BLUE(), RGB() */
5:
6: unsigned short pixel[ 512 ];
7:
8: void main()
9: {
10:     unsigned int x, y;
11:     unsigned short c;
12:     unsigned int r0, g0, b0;
13:     int r1, g1, b1;
14:
15:     if ( CRTMOD(-1)!=12 ) {
16:         fprintf( stderr, "画面が 65536 色モードになっていません。%n" );
17:         fprintf( stderr, "65536 色モードにするには、SCREEN 1,3,1 とします。%n" );
18:         fprintf( stderr, "そのうえで画像をロードしておいて実行してください。%n" );
19:         fprintf( stderr, "画像の色を強調します。%n" );
20:         return;
21:     }
22:     for ( y=0; y<512; y++ ) {
23:         get( 0, y, 511, y, pixel, 512*sizeof(short) );
24:         for ( x=0; x<512; x++ ) {
25:             c=pixel[x];
26:             r0=RED(c);
27:             g0=GREEN(c);
28:             b0=BLUE(c);
29:             if ((r1=(r0*2)-(g0/2)-(b0/2))>31) r1=31; else if (r1<0) r1=0;
30:             if ((g1=(r0/2)+(g0*2)-(b0/2))>31) g1=31; else if (g1<0) g1=0;
31:             if ((b1=(r0/2)-(g0/2)+(b0*2))>31) b1=31; else if (b1<0) b1=0;
32:             pixel[x]=RGB(r1, g1, b1);
33:         }
34:         put( 0, y, 511, y, pixel, 512*sizeof(short) );
35:     }
36:     return;
37: }
```


多数のソースファイルを管理する

XCにMAKEが付いてきた

Nakamori Akira 中森 章

C言語では、プログラムのモジュール化によって大きなプログラムも効率的に作成されます。が、そのためには数々のファイルをきちっと保守管理しなくてはなりません。そこで登場のMAKE。XCの環境がまた一歩進みました。

C言語で巨大なプログラムを作成するときに必要になるもの。それは努力と忍耐、そしてMAKEです。

プログラム開発の基本は、プログラムをモジュール化して別々のファイルで作成し、最後に結合するのが一般的です。これがいわゆる分割コンパイルというやつですが、これはファイルの数が多いと結構煩わしい作業になります。たとえば、あるファイルを修正したり変更したりしたとき、その修正や変更がほかのファイルのプログラムにも影響を及ぼすものであるなら、影響されるファイルをすべて再コンパイルしなければなりません。多数のファイル間の依存関係なんて開発期間がちよっと長くなると忘れてしまいますからね。

そこでMAKEが必要になります。ファイル間の依存関係を気にせず自由にファイルを修正/変更する勇気を与えてくれるツールがMAKEなのです。XCのver.1ではコンパイルオプション (/M) でMAKEもどきなコンパイルを行うこともできましたが貧弱な機能しか持っていませんでした。ところが、XCのver.2.0ではとうとうMAKEが標準で付属するようになったのです。たかがコンパイラのおまけと侮ってはいけません。プログラマーズマニュアルのMAKEの解説を読めばそれがかなり本格的なものであるとわかります (馴染みのない人には理解できないかも)。

今回はMAKEの解説ということなのですが、豊富過ぎる機能をすべて紹介するのは不可能ですから、MAKEの初歩の初歩を説明してその有用性について知ってもらう程度にとどめましょう。

MAKEが必要なそのわけは

MAKEの機能をひと言でいえば、指定した手順に従って複数のソースファイルをコンパイルしたりリンクすることです。このときMAKEはあらかじめ与えられているソースプログラム間の依存関係を調べ、

変更したソースファイルが影響を与える最小限のファイルだけをコンパイルしたりリンクしたりしてくれるのです。

たとえば、次のような例を考えてみましょう。プログラムprog.xは、Cのソースファイルmain.c, subr1.c, subr2.cをコンパイルして作られていると仮定します。さらにsubr1.cとsubr2.cは共通のヘッダファイルcommon.hをインクルードしているものとします。このときprog.xはXCでは、

```
cc /Fxprog main.c subr1.c subr2.c
```

を実行することによって作られますが、このやり方は、まだソースファイルのバグが取り切れてない状態では、どれかのソースファイルを変更するたびに3つのソースファイルをすべてコンパイルすることになって効率がよくありません。

コンパイル時間を節約するためには、

```
cc /Fc main.c
cc /Fc subr1.c
cc /Fc subr2.c
```

というように/Fcオプションによって、すべてのソースファイルをリンクする直前のオブジェクトファイルの形式 (拡張子が.oであるファイル) で保存しておき、その後、

```
cc /Fxprog main.o subr1.o subr2.o
```

によってオブジェクトファイルをリンクしprog.xを作成します。

もし、subr1.cに変更があった場合は、

```
cc /Fc subr1.c
```

というように、変更のあったソースファイルのみをコンパイルしてオブジェクトファイル (subr1.o) を作り、再び全体をリンクすれば、新しいprog.xを作ることができます。この場合、変更のなかった2つのソースファイルをコンパイルする必要はありません。

次はcommon.hが変更された場合を考えましょう。これは、そのファイルをインクルードしているsubr1.cとsubr2.cの2つが変更されたのと同じことになります。したがって新しいprog.xを作るためには、

```
cc /Fc subr1.c
```

```
cc /Fc subr2.c
```

によってsubr1.cとsubr2.cをコンパイルして新しいsubr1.oとsubr2.oを作り、

```
cc /Fxprog main.o subr1.o subr2.o
```

によってリンクしなければなりません。これはひとつのファイルの変更が2つ以上のソースファイルに影響を与える例です。もし、subr2.cがcommon.hをインクルードしているのを忘れて、subr1.cのみしかコンパイルしなかったらリンク後のprog.xがまともに動くわけはありませんね。分割コンパイルを行うとこのようにソースファイル間の依存関係にいつも注意しなければならないのです。

上の例ではファイルの数が少ないので、頭の中で少し考えればある変更に対してどのソースファイルをコンパイルすべきかということはいくらでもわかります。しかし、ファイルの数が多くなるとこんなにすんなりとはいきません。なによりも、変更された多くのソースファイルをコンパイルするためのキーボードからのコマンド入力はいくらでも面倒です。

MAKEはこのような煩わしい手順から解放してくれるツールなのです。

ところで、定型的なコンパイル、リンクという作業を一括して行うことはHuman 68kのバッチファイルを使っても可能です。しかし、バッチファイルではソースファイル間の依存関係を調べて必要なものだけをコンパイルするなどという芸当は不可能に近く、仮にユーティリティプログラムなどを駆使して実現できたとしても非常に複雑なものになってしまうでしょう。

簡単な手順でコンパイルやリンクを自動化することがMAKEの意義なのです。

注意) ここで説明したオプションはXC ver.2.0のもの。/FcはXCのver.1では/L、GCCでは-cである。また、/FxはXCのver.1では/Z、GCCでは-oである。

便利さの秘密は

MAKEは変更されたファイルが影響を与えるソースファイルを自動的に探し出し

ます。といっても、MAKEは単なるユーティリティプログラムにすぎませんからそこにはトリックがあります。それは、それぞれのファイル間の関係やファイルが変更されたときの動作を記述するMakefileというファイルです（もともとファイル名の太文字、小文字の区別はないのでmakefileでもMakeFileでもなんでもよい）。

MAKEはこのMakefileの内容を頼りにMAKE自身が行うべき動作を決定します。また、ファイルが変更されたかどうかの判断はコンパイルなどによって作られるファイルの作られた日時がその元になるソースファイルの作られた日時より前かどうかによって行います。したがってMAKEを使用する場合はあらかじめMakefileを書いておく必要があります。

それではMakefileの書き方について説

リスト1

```
1: prog.x : main.o subrl.o subr2.o
2:      cc /Fxp prog main.o subrl.o subr2.o
3: main.o : main.c
4:      cc /Fc main.c
5: subrl.o : subrl.c common.h
6:      cc /Fc subrl.c
7: subr2.o : subr2.c common.h
8:      cc /Fc subr2.c
```

リスト2

```
1: /*
2:      main.c   いわゆるメインプログラム
3: */
4: main()
5: {
6:      int x;
7:      scanf("%d", &x);
8:      printf("func1=%d\n", func1(x));
9:      printf("func2=%d\n", func2(x));
10: }
```

リスト3

```
1: /*
2:      subrl.c
3: */
4: #include "common.h"
5:
6: func1(x)
7: int x;
8: {
9:      return( CONST*x );
10: }
```

リスト4

```
1: /*
2:      subr2.c
3: */
4: #include "common.h"
5:
6: func2(x)
7: int x;
8: {
9:      return( CONST*x );
10: }
```

リスト5

```
1: /*
2:      common.h   たったこれだけです・・・
3: */
4: #define CONST 10
```

明しましょう。Makefileでは基本的には、ソースファイルとそれをコンパイルやリンクによって作られるファイル（ターゲットファイルという）、およびソースファイルからターゲットファイルを作るためのコマンド行を記述します。具体的には、

ターゲットファイル：ソースファイル
コマンド行

という記述です。ひとつのMakefileの中にはこの組み合わせが必要な数だけ記述されています。ここで、:（コロン）がファイルの依存関係を示しています。:の右側にあるソースファイルはいくつあってもかまいません。ここに必要なソースファイルを書き忘れると当然MAKEは正しく動きません。コマンド行には通常ソースファイルからターゲットファイルを作るためのコマンドを書きます（コマンド行はなぜかタブで始まらなければならない：重要）。コマンド行は続けて何行書いてもかまいません。このとき各コマンドが連続して実行されます。

これだけの知識でMakefileを書くことができます。たとえば、先に述べたmain.c, subrl.c, subr2.cからprog.xを作る場合にはリスト1に示すようなMakefileを書けばよいでしょう。最終的に作られるファイルの関係をいちばん最初に書くことを除けば、あとは適当な順序で記述してもなんとか動くでしょう。結構簡単だと思いませんか？ Makefileを書いたあとは、キーボードから、

MAKE

と打ち込むだけで、すべてのことをMAKEがやってくれるでしょう。

ところで、リスト1のMakefileは単純明快ですが、みなさんが目にするMakefile、たとえばPDSなどをコンパイルするときに使用するMakefileはもっと複雑でわけのわからないものが多いと思います。これはMAKEの提供する省略機能やマクロを駆使してMakefileが効率よく（読みにくいけどカッコイイ）書かれているためです。どのように記述しようと効果は同じ

ですが、Makefileをかつこよく書きたい人はマニュアルで勉強してくださいね。

注意）コンパイルオプションの差異から、リスト1のMakefileはXC ver.2.0でしか正しく動かない。XCのver.1やGCCで使用する場合はコンパイルオプションを変更する必要がある。また、この理由からUNIXで使用されているMakefileを持ってきても、そのままではまづ正しく動かない。

MAKEを体験しよう

説明だけではおもしろくないので実際にMAKEを動かしてみましょう。Human68kのMAKEはMAKE.Xというファイルです。これをPATHの通じているどこかのディレクトリに置いておいてください。これが最小限の準備です。次にMakefileが必要ですが、これはリスト1のものを使いましょう。これに従って3つのソースファイルとひとつのヘッダファイルを用意します。これは何でもいいのですが、とりあえずリスト2～5を使うことにしましょう。リスト2からリスト5のファイルとリスト1のMakefileをひとつのディレクトリに集めたら、

MAKE

と打ち込んでください。main.c, subrl.c, subr2.cが次々とコンパイルされリンクされてprog.xが作成される様子を見ることができます。これを確認したら、

- 1) subrl.cだけを書き換えたあとにMAKEを実行する（MAKEと打ち込む）。
- 2) common.hを書き換えたあとにMAKEを実行する。

と操作をしてみてください。1)の例ではsubrl.cだけがコンパイルされなおすこと、2)の例ではsubrl.cとsubr2.cのみがコンパイルされなおすことが確認できます。

注意）XCのver.1とver.2.0では標準的なヘッダファイルであるFEFUNG.Hに互換性がないので注意。FEFUNG.Hとコンパイラのバージョンが一致しないと正常にコンパイルできない。

*

現在、UNIX上でのプログラム開発においてMAKEを使用することは半ば常識のようになっています。今回の解説でMAKEに興味を持った人は、是非とも実際にMAKEを使用してみることをお勧めします。特にソースプログラムが2つ以上になる場合は、今回紹介したごく初歩的な機能だけでも結構重宝するはずですよ。

それにしても、ソースコードデバッガも付属したことだし、XCもいよいよまともにプログラミングできる環境を目指してきただんだというのが最近の実感です（バグもまだあるようだけど）。

MIDI制御が加わった 新しい音楽ドライバOPMDRV2.X

Nishikawa Zenji 西川 善司

新しいCコンパイラとともに突然現れたOPMDRV2.X。初めてのシャープ提供によるMIDIドライバの概要と使い方を見ていきましょう。MT-32を使用したサンプルプログラムも掲載します。参考にしてください。

私が駅前の新興宗教の勧誘によく捕まるモゲランチョ西川善司です。突然ですが、「C compiler PRO-68K ver. 2.0」を買ったとバージョンアップされた音源ドライバ「OPMDRV2.X」がついてきます。ここでは、そのドライバが前の「OPMDRV.X」とどう違うのか、またマニュアルに載っているサンプルプログラムでは少しわかりにくいと思われるチャンネルアサイン関係について説明します。

拡張された命令たち

「OPMDRV2.X」になって拡張されたのはMIDI出力関係です。はっきりいってしまえば、FM音源関係についてはなにひとつ変わっていません。まあ、以前のミュージックデータとの互換性のからみのため、変えようがなかったのでしょう（期待した人残念でした）。

その代わりといっってはなんですが、MIDI制御用の新しいコマンドが追加されているのでそれをリストアップしてみます。

```
md_play
md_stop
md_cont
md_init
md_stat
md_on
md_off
md_regr
md_rdgw
md_wrt
```

表1 MIDI拡張MML一覧

MMLデータ	意 味	パラメータ範囲
' (アポストロフィ)	MIDI拡張MMLの使用を開始/終了する	
Tn	MIDI送信チャンネルセット	1~16
Pn	MIDIプログラムチェンジ	1~128
On, m	ノートオン	0~127
Fn	ノートオフ	0~127
\$n	ダイレクト送信データ	0~255
n	ダイレクト送信データ	0~255

（前記以外に既存命令と同名でMIDIに対応している、というのもあります）。

まず、上から5つは説明するまでもなく既存の命令「m_play」、「m_init」など……のMIDI版といった感じです。その下の「md_on」「md_off」は引数で与えたMIDIチャンネルへの出力をスイッチする命令です。そのまた下の2つ「md_regr」「md_rdgw」はX68000のMIDIコントローラ、YM3802からのレジスタ読み込み、またはレジスタへの書き込み命令です（ん〜、マニアック！）。最後の「md_wrt」はMIDIの生データの出力命令で、まあ、これさえあれば理論上はなんでもできるというやつですな。

拡張されたMML

MIDI楽器をMMLでコントロールできるようにと「拡張MML」なるものが使用可能になりました。「拡張MML」は、従来の「OPMDRV.X」のミュージックデータと区別しやすくするためか「'」（アポストロフィ：[SHIFT]+7）でくくられた中に記述します。既存の「FM音源専用のミュージックプログラム」をMIDI対応に改造するには任意の位置にこのアポストロフィによってくくられた拡張MMLを挿入すればよいわけです（見た目で拡張MMLが使用されているのがわかりますね）。

表1を見てください。これらが拡張MMLです。注意したいのは拡張MMLはいままでのMMLと違って横にずら一つと並べて書くことができず、各コマンド間を必ず「,」カンマで区切らなければいけない点です。たとえば、あるMMLトラックをMIDIチャンネル1に送ることにし、音色番号99に切り替えるには、
'T1, P99'
としなくてはなりません。しか

し生データなどを送る際には特にコマンドもいらずアポストロフィ「'」の中にデータ数値を書き、同じ要領でそれをカンマで区切るだけでよいので、慣れてしまえばそんなに違和感はなくなります。たとえばMT-32のMIDIチャンネル2のベンドを基準値に直す場合を考えてみますと、

'T2, 225, 0, 64'

というふうになります。表1を見ると「ノートオン」などの命令もありますが、音階のMMLはちゃんと使えるので、これらのお世話になることはほとんどないと思います。

ところで、表1を見て「ちょっと命令が少ないんじゃない？」と感じた方も多くいることでしょう。確かにボリュームコマンドがない、ベンドがない、パンがない、ましてコントロールチェンジもない……。ボリュームについては「ベロシティ」をMMLの「V」でコントロールできますがマスターアウトプットのボリュームがないのはどっちにしろかなりの痛手です。まあ、先ほどの例の「ベンド」にしても3バイトデータをいちいち送らなくてはいけないので、MIDIデータの勉強にはなりますよ。

使用して気づいたことなど

まず、使用して戸惑ったのが「m_assign」命令。おそらく私と同じように「音が鳴りましえーん」と泣きそうになってしまう人も出てくると思うので、これについて少し説明しておきます。

この命令は周知のとおり、

m_assign (CH, TR)

のように音源チャンネルを「m_alloc」で確保したMMLトラックに割り当ててるものです。「OPMDRV2.X」ではCHの部分は前と同じく1~8がFM音源です。MIDI楽器（つまりMIDIチャンネル）は便宜的に9~24となっていますので、たとえばMIDIチャンネル1~8をトラックバッファ番号1~8にアサインするには、

for i=1 to 8

のようになくしてはいけません。たとえばとで拡張MMLの「T」コマンドを使って初めにアサインしたのとは違うチャンネルを使うとしてもです。よく意味のわからない人はMIDI楽器のときは「i + 8」のようにするんだと覚えておけばいいでしょう。

のようにします。また「md__stop」や「md__cont」といった演奏制御命令も同様の引数をとります。ですから、

OPMDRW、Xと同じ内部処理のまま24トラック化されているようで、MIDIを使うと処理が重くなりデータによってはテンポずれを起こしてしまいます（FM音源だけ使うには困らないのですが）。

MIDI楽器のマニュアルを見れば載っていることですが一応プログラムで使用しているMIDI数値データを解説します。

で文字変数の内容をMT-32の画面に出力します。ほとんどお遊びですがエクスクルーシブメッセージの勉強になると思いますので初心者も参考にしてください。

©システムサコム

```

4 c8.cc4e8.ee4 d-8.d-d-4e-8.e-e-† e8.ee4d8.dd4 d-8.d-d-4e-8.e-e-
-4 c8.cc4e8.ee4 d-1e8.ee-4r4
400 m_trk(4,"t4" q6 o4 v13)+a)
410 m_trk(4,"[do]" +b+"[loop]")
420 /*
430 /* B A S S
440 /*
450 a="L16e8.ee8<e8>e8.ee8<e8>"
460 b=":|:e8.ee8<e8>a-8.a-a-8<g8> g-8.g-g-8<g-8>b8.bb8<e-8>|
:a8.aa8<a8>b8.bb8<b8> a-8.a-a-8<a-8d-8.d-d-d-8<d-8> |la8.aa8<a8>b
8.bb8<b8> e8.ee8<e8>e8.ee8<e8> :|a1b8.bb8<b8>b8.)b8b<d-e-
470 m_trk(5,"t5,p72,180,7,127,180,10,63,228,0,64" q8 o2 v14
+ a)
480 m_trk(5,"[do]" +b+"[loop]")
490 /*
500 /* D R U M S
510 /*
520 a="L16f+8.f+f+8a+8f+8.f+f+8a+a+
530 b=":|:10f+8.f+f+8a+8f+8.f+f+8a+a+:|f+4a+4a+4a+4f+8.f+f+8a+a+
f+4rrf+f+
540 m_trk(6,"t10,p50,185,7,127,233,0,64" q8 o2 v8)+a)
550 m_trk(6,"[do]" +b+"[loop]")
560 /*
570 a="L16c8.cc8c8c8.cc8c8
580 b=":|:10c8.cc8c8c8.cc8c8:|c1c4c1c1c8.cc8d8c4rr<c>f
590 m_trk(7,"t10" q8 o2 v10)+a)
600 m_trk(7,"[do]" +b+"[loop]")
610 md_play(<HFFFFFF00):end
620 func mt_print(m:str)
630 dim char d(7):={&H0F,&H41,&H10,&H16,&H12,&H20,&H0,&H0}
/*command and address data
640 int i,s
650 char a[4],l
660 if len(m)>20 then print"メッセージが長すぎるたよ…。こんな
のを送るなんてオラには出来ねえ"return() /* ERROR
670 for i=0 to 7:md_wrt(d[i]):next
680 s=0:l=len(m)
690 for i=1 to l
700 a=asc(mid$(m,i,1))
710 md_wrt(a)
720 s=s+a
730 next
740 for i=5 to 7
750 s=s+d(i)
760 next
770 e=128-(s and &HFF):e=e and 127
780 md_wrt(e) /*send checksum data
790 md_wrt(&HF7) /*end of exclusive
800 endfunc

```


BASICコンパイラ派に贈る便利ソフト

XBASStoC CHECKER

Nishikawa Zenji 西川 善司

X-BASIC でプログラムを作成しXC (またはGCC) でコンパイル。これが意外と効率のよい開発方法なのです。ここではBASICユーザーを支援する強力なツールを紹介します。これまでコンパイルできなかったプログラムもこれで大丈夫?

いまどき、C言語が使えないパソコンはあまりない。では、どこで選ぶか?

お答えします。それは、BASICもコンパイルできる「C compiler PRO-68K」が動くX68000です。

目のつけどころがシャープでしょ。

と、いうわけで若葉マークのX68000ユーザーのなかにはこの事実を知らなかった人もいんじゃないかな。そう、「C compiler PRO-68K」(以下XC)はX-BASICのプログラムをコンパイルすることができるんです。

具体的にはまずBASICのソースリストをCのソースに変換し、これをコンパイルし機械語の実行ファイルを作成します。これは大変便利な機能なんだけれど、BASICならなんの問題もなしに動いていたものが、いざコンパイルしてみると動作がちよっと変、なんてことがよくあるのです。

今回発売となった「XBASStoC CHECKER」(以下チェッカ)はこういったX-BASIC→Cの変換段階でプログラマの目的と違って変換されてしまう部分をわかりやすく指摘してくれるソフトなのです。

チェッカの概要

さて、実際にこのソフトは大きく分けて2つの動作モードがあります。



X68000用 5"2HD版 9,800円(税別)
シャープ 03(260)1161



写真1 起動時の画面

ひとつはグラフィカルモード。これは、すべての操作をビジュアルに行えるもので画面写真のようなメニュー画面をマウスで次々にクリックしていくことにより作業が進められます。こちらのモードで動作させるにはG-RAM (グラフィックRAM) をRAMディスクとして使用していないことが大前提です。理由は見てのとおり、メニューをグラフィックで描いているためです。

もうひとつは一般的なコマンドラインで動作させるモードです。まあ、通常の外部コマンドのように動作するモード、と考えていただければ結構でしょう。ですから、バッチファイルなどを使った自動処理にはこちらが向いていますね。

動作環境における注意がいくつかあるのでこれを挙げておきます。まず、両モードとも動作にはテンポラリディスク (まあ、コンピュータがメモ帳代わりにディスクを使うと考えてください) が必要です。チェッカは動作にあたっていくつかの中間ファイルを作成しますので、ある程度フリーエリアのあるディスク上で作業するか、または、

A>TEMP ? :

を実行して十分空きのあるディスク (RAMディスク/ハードディスクでも可) をテンポラリとして指定する必要があります。

また、両モードとも浮動小数点演算パッケージが、またグラフィカルモードではFM音源ドライバをデバイスドライバとして登録しておく必要があります。

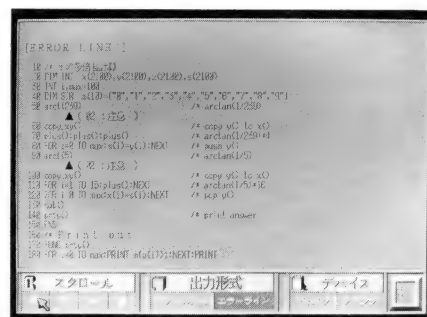


写真2 エラーレポート

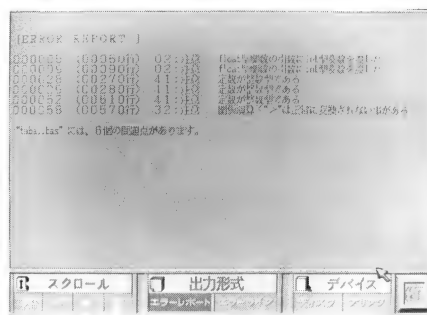


写真3 ここがエラー

作業手順はこんな感じ

さて、実際にリストをチェッカにかけてみることにしましょう。ここでは、グラフィカルモードを使用して話を進めます。

立ち上げるとタイトルのあと、写真1のようなファイル選択画面となります。ここでチェックしたいプログラムをマウスで選択します。ちなみに拡張子が「.BAS」以外のファイルは表示されませんのであしからず。

さて、画面右上の「エラーレポート」と「エラーライン」ですが、これはチェッカがユーザーにどんな情報を提供すればいいかの指定をするものです。「エラーレポート」ではエラーの概要とその行番号を表示してくれます。「エラーライン」では実際にソースリストのどの部分にトラブルがあるのかを指摘してくれます。まあ、通常は両方のスイッチをオンにしておくといでしょうね。

その下の「デバイス」というのは先ほどスイッチ指定した「エラー情報」をどこへ出力するかを決定するものです。「画面」と「プリンタ」は読んで字のごとくですが、「ディスク」では「エラー情報」をファイルに書き出してくれます。

さあ、あとは「実行」で実際にチェックしてみましょう。

チェック終了メッセージのあと、初めての出力形式の設定のところで「エラーレポート」をオンにしてあれば写真2のような画面が表示されることでしょう。「エラーライン」のみをオンにしてあると写真3のような画面が出てきます。また、両方のスイッチをオンにしてあれば画面下の「出力形式」のメニューをマウスでクリックすることによって2つの「エラー情報」を切り替えることができます。

この時点で「デバイス」を変更し「エラー情報」をディスクやプリンタに出力することも可能なので、

- ・初めは「デバイス」を「画面」に設定しておく

- ・必要ならば「ディスク」や「プリンタ」に出力し保存する

というのが一般的な使い方でしょうか。

さあ、エラーを確認したあとは、BASICやエディタに帰り、サクサクとデバッグをすればいいわけです。

はふ、謎が解けた！

このソフトのマニュアルの後半にはチェックの出力する「エラー情報」の詳細とその対応策が載っていますが、これは大変参考になりますぞ。ここには「原因」という項目があるのですが、これを読めば「XC」がX-BASICのプログラムをどう変換してしまうのか、また、C言語とBASICの相違点などを知ることができるのです。

そうですね、たとえば、ゲームなどのキー入力の処理なんかで、

$X = X + (A = "4") - (A = "6")$

ということをやりますが、BASICでは正常に動作しても、これをCに変換した場合には不都合が生じます。これは、BASICとC言語とでは論理演算の出力結果が違うために起こるものです。具体的には、

BASIC	真： -1
	偽： 0
C	真： 1
	偽： 0

です。ですから上の例をコンパイルするときには「+」と「-」を入れ替える必要が

あるでしょう。

このように長い間原因不明だったバグの正体が、ここを読むにつれてあれよあれよと解明されていく気分はさながら推理小説の後半を読んでいるようですよ。

まあ、これは私ひとりの願望かもしれませんが、わかりきった「エラー」もしくは「注意事項」は自動的に直してくれる機能がほしかったですね。たとえば文字列の最後を、

A = " ABCDEFGHIJKLMNOP"
のように「」でくくっていないとチェックは「注意」を促してくるのですが、自動的に「」をつけ足してくれるくらいの気の利いた処理をしてもバチは当たらないと思うのですが。まあ、勝手にやられると気持ち悪いという人もいるでしょうからスイッチ指定できるようにするとか、ね。

まとめ、とその他気づいた点

ゲーム誌「LOGIN」なんかの「ソフコン」の入賞ソフトにはX68000用のものが比較的多いですが、その中にはX-BASICからコンパイルしたものがよく見受けられます（「C ON Z」とかありましたね）。また、多

くの同人ソフトのグループなんかもこの「XC」のBASICコンパイル機能を愛用しているようです。まさに、今回の「XBASStoC CHECKER」はこういう人たちにおすすめです。しかし、こんな便利なもの、「XC」の本体につければよかったのにねえ。

最後に気づいた点をいくつか。

グラフィカルモードは大変便利ですが、G-RAMを使用しているのはちと痛いですね。というのは、RAMを何Mバイトも増設している人はともかく、多くの人はG-RAMをテンポラリディスクにあてていると思うのです。それなのに、G-RAMをRAMディスクにしていると使用不可能というのは痛い。だいたい、あんな派手な画面にする必要はないでしょうに。「XC v. 2.0」のソースコードデバッガや「COMMANd, X」用外部コマンドの「FORMAT, X (v. 2.00以降)」「SWITCH, X (v. 2.00以降)」を見てもわかるようにテキスト画面だけでも十分見やすい画面は作れるのです。

いずれにせよ、大変便利なことは確か。まだ、機械語はわからないけどゲームなんかを作ってみたい、なんて思っている人は「XC」とセットで買うといいかもよ。

図 マニュアルはいろいろ参考になる

28. 注意「int関数は実行結果が異なる可能性がある」

原因 int関数に渡されたパラメータ（数値・変数）がマイナス値の場合、実行結果がおかしくなることがあります。
例えば、「print int(-3.3#)」を実行した場合、インタプリタでは「-4」、コンパイラでは「-3」という結果が返ってきます。

解決策 「修正例」で示すように、最大整数値（int）の算出はユーザー関数の方で行うようにします。
標準関数「int()」は、なるべく使用しないようにします。

問題例 100 print int(-2.8#)
▲（28：注意）

修正例 次のようにユーザー側で新しいint関数を作成し（「int2()」）、使用するようにします。インタプリタのときでも、「int()」はなるべく使わないようにしてください。

```
100 print int2(-2.8#)
110 end
10000 func int int2(a;float) /* ユーザー int 関数
10010 float b
10020 b=fix(a)
10030 if a<b then b=b-1
10040 return(b)
10050 endfunc
```


●STACKコンパイラ登場

インタプリタ言語STACKにコンパイラ版ができました。STACKとフルコンパチですので、インタプリタで開発、テストラン、そしてコンパイルして実行と、より手軽にアセンブラによらない高速アプリケーション開発が扱えます。

作者はもうお馴染みの平井真二氏です。

もともと、S-OSオリジナル言語のなかでも、一風変わった風貌を持つインタプリタとコンパイラシステム。スタック型という考え方に慣れさえすれば、常用の言語としてアセンブラを使うでもない分野（つまりほとんどの処理）をこなすことができるでしょう。実用面でももちろんマルの性能です。

手軽さのなかには、全体の小ささとあいまって「コンパイルが高速」だということも含まれます。インタプリタとあわせて使用するのですが、コンパイルが高速なら、コンパイラの実在を感じさせないシステムを組み上げることも可能。

S-OS用のオリジナル言語というよりはSLANGの優秀さが目につきますが、インタプリタ/コンパイラ環境はやはり捨てがたいものがあります。SLANGインタプリタなんてできないかな……。

第101部

STACKコンパイラ

●Cは延期

予告していたC言語の移植はちょっと延期となりました。期待していた方ごめんなさい。

豊富なラインアップを誇るS-OSの言語処理系ですが、なぜか欠けていたのがC言語でした。S-OSスタート当時は「噂のC言語を使ってみよう」という要望が多かったものです。伝説の処理系だったC言語も、いまやアセンブラ代わりの開発言語として日常的に定着した観があります。

これまでに何度かC言語をサポートしようという気運が盛り上がったことはあったのですが、いずれも実現しなかったのは「処理系だけなら作りますよ。ライブラリのほうはお願いしますね」というスタッフの弁に代表されるように、処理系作りの難しさよりライブラリを揃えることの面倒臭さが原因となっていたのではないのでしょうか。

C言語を完全にアセンブラ代わりに使うのならともかく、高級言語的に使用するのならライブラリが充実していなければ手も足も出ません。C言語を使いたいと要望していた皆さんのなかには、アセンブラは無理でもCならS-OS用のアプリケーションを作れそうだという方が多く、ライブラリなしのC言語など考えられなかったのです。

ですから、CP/M上のC言語を持ってこようというのは、むしろ自然な発想といえるでしょう。

ただし、誌面に掲載できるのは変更箇所と手順に限られますから、実際にコンパイラのインプリメントを行うのは非常に面倒な作業となりそうです。少なくとも、CP/M、MACRO-80ほか、これまで準備してきたCP/Mファイルコンバータ、やWZD、WLK、WLBなどのシリーズがすべて必要となります。覚悟だけはしておいてください。

●S-OSの系譜 (16)

マシン語ファイルの共通化を果たしたS-OSと、グラフィックデータの共通化を果たした高機能グラフィックパッケージMAGIC。両者の能力を融合するとどんなものができるのか。この試みは1987年3月号で行われました。掲載されたプログラムはMAGE（メイジ）と命名されました。MAGEとは魔法使いの意。MAGIC（魔法）を使いこなすMAGEの登場です。

MAGEはピクチャーエディタとストーリーエディタの2つのプログラムから成っています。ピクチャーエディタで線画の絵を描いておき、それをバラバラ漫画のように順次表示することによってアニメーションしようというのです。作成された絵はMAGICのデータですからMAGICが移植されたMZ-1500/2000/2200/2500、XI/turbo、PC-8801シリーズ、そしてSMC-777で共通に扱うことができます。もちろんMAGE自身はS-OSのアプリケーションですからこれら全機種で共通のプログラム。グラフィックを使ったアプリケーションでプログラムもグラフィックデータも共通という大きなイベントが繰り広げられたのです。

DōGAプロジェクトの出現したいまとなつては、思わず「線画？」と問いかけたくなるかもしれませんが、64Kバイトのメモリではフルアニメーションなど無理なことから切り捨てた結果、線しか表現できないことがあってカリグラフのような独特の魅力をもたらしていたものです。ピクチャーエディタで描いた絵はストーリーエディタによって管理され、タイムテーブルに従って順次表示されていきます。いかに少ない絵で効果的に見せるか、手腕の問われるところでした。

MAGICは続く1987年4月号でMZ-80B/B2にも移植されました。MZ-80B/B2のグラフィックは320×200。しかもG-RAMは2プレーンしかありませんが、解像度の違いは内部で640ドット→320ドットの変換を行ってデータの共通化を実現しています。もちろんMAGEにも対応。MZ-80B/B2ユーザーからのこの投稿プログラムはユーザーの底力といったものを見せつけてくれました。

STACKコンパイラ

Hirai Shinji

平井 真二

FORTHとBASICを足したようなスタック型言語STACKがコンパイラになりました。インタプリタ上のプログラムがこれまで以上の速度で実行できます。使用の際にはSTACKインタプリタが必要です。

STACKコンパイラはSTACKフルコンパチのコンパイラです。ランタイムルーチンを含めても4Kバイト弱とコンパクトですが、これはSTACKのセミコンパイル結果を利用しているからです。そのため本コンパイラを利用するにはSTACKが必要です。速度が気になるとは思いますが、STACKのセミコンパイラと比べると1.5～2.5倍ほど速くなります。

入力&実行方法

まず、リスト1のダンプリストをMAC INTO-Cなどのマシン語入力ツールから打ち込んで、実行アドレス3A13_Hでセーブしてください。

使い方は、まずコンパイルしたいプログラムをSTACKのCコマンドでセミコンパイルしてください。次にS-OSのモニタに戻り、STACKコンパイラをロードして、

#J3A13

と入力して起動してください。

コンパイラは次のような質問をしてくるので、16進4桁で答えてください。

TEXT ADDRESS :

セミコンパイル結果の格納先頭番地。

OBJECT ADDRESS:

オブジェクトの発生開始番地 (セミコンパイル結果と重ねることはできない)。

VARIABLE TOP :

変数領域の先頭番地。

STACK TOP :

パラメータスタックの先頭番地。

RET_STACK TOP :

リターンスタックの先頭番地。

OFFSET ADDRESS:

オフセット。通常は0000とします。

以上でコンパイルが開始されます。しばらくすると、オブジェクトの終了アドレスを表示してS-OSに戻ります。なお、このアドレスはオフセットを含めた値なので注意してください。

コンパイルが終了したら、ランタイムルーチン (3000_H～3A12_H) とオブジェクトを含めた範囲で実行アドレスをオブジェクトの先頭番地としてセーブしてください。

#Jオブジェクトの先頭番地でコンパイルしたプログラムが動きます。

サンプルプログラム

STACKコンパイラの動作をチェックするためのサンプルプログラムを掲載します。エラトステネスのふるいを用いた素数を求めるプログラムです。

まず、STACKインタプリタのエディタで入力し、

]C6000

でセミコンパイルしてください。

次にモニタに戻り、STACKコンパイラを起動し、

TEXT ADDRESS : 6000

OBJECT ADDRESS: 4000

VARIABLE TOP : 4200

STACK TOP : 5000

RET_STACK TOP : 5800

OFFSET ADDRESS: 0000

と入力すると、OBJECT ENDのアドレスを表示して終了します。実行はS-OSのモニタからJ4000です。

リスト1 STACKコンパイラ

```
3000 00 00 00 AE 00 00 00 00 : AE
3003 00 00 00 00 00 32 08 30 : 6A
3010 ED 53 09 30 22 08 30 C9 : 9F
3018 DD E1 D1 E1 19 E5 DD E9 : 34
3020 DD E1 D1 E1 B7 ED 52 E5 : 4B
3028 DD E9 DD E1 D1 E1 CD B0 : B3
3030 30 E5 DD E9 DD E1 D1 E1 : 4B
3038 CD C3 30 E5 DD E9 DD E1 : 29
```

```
3040 D1 E1 CD C3 30 D5 DD E9 : 0D
3048 DD E1 D1 E1 CD C3 30 E5 : 15
3050 D5 DD E9 DD E1 D1 B7 : C2
3058 ED 52 21 00 00 20 01 23 : A4
3060 E5 DD E9 DD E1 D1 B7 : D2
3068 ED 52 21 01 00 38 01 2B : C5
3070 E5 DD E9 DD E1 D1 18 : 33
3078 EE DD E1 E1 D1 B7 ED 52 : 54
```

サンプル

```
1 ; エラトステネス / フルイ
2 8190 .M
3 $8000 .F
4 ;
5 #F #M 0 FILL ; CLEAR
6 2 PRINT
7 3 .I
8 ;
9 %1
10 #M #I < IF 1 BELL END
11 #F #I + PEEKB =0 IF #I COPY
PRINT GOSUB 2
12 INC I INC I
13 GOTO 1
14 ;
15 %2
16 .B
17 2 #M #B / DO
18 1 #F #B I? * + POKEB
19 LOOP! RET
```

最後に

コンパイルはかなり安易な方法で行ってありますが、文法解析およびエラーチェックがセミコンパイル時に終わっているためコンパイル速度はそれほど遅くはありません。

ランタイムルーチンは3000_H～3A12_Hに固定されていますがそれほど問題はないでしょう。セミコンパイラと比べ、それほど速くはありませんが、実行時にインタプリタがいらない、512以上のラベルを使っても速度が落ちない、マシン語サブルーチン化ができるなどのメリットがあります。普通に使う分にはセミコンパイラで十分だと思いますが、もっとスピードがほしいというときにはぜひ本コンパイラを使ってみてください。6月号のSQUASH!も見違えるように操作性がよくなります。

今後は、またまた時代の流れに逆らって2～3Kバイト程度の記号型言語でも作ろうと思っています。

SUM: 96 80 11 6C EE EF 4 61 2D 2596

```
3080 21 01 00 20 01 2B E5 DD : 30
3088 E9 DD E1 D1 E1 7D A3 6F : E8
3090 7C A2 67 E5 DD E9 DD E1 : EE
3098 D1 E1 7D B3 6F 7C B2 67 : E6
30A0 E5 DD E9 DD E1 D1 E1 7D : 98
```


30A8 AB 6F 7C AA 67 E5 DD E9 : 52
30B0 4D 44 3E 10 21 00 00 29 : 29
30B8 CB 23 CB 12 30 01 09 3D : 42
30C0 20 F5 C9 4B 42 54 3E : 5A
30C8 10 21 00 00 CB 23 CB 12 : FC
30D0 ED 6A E5 B7 ED 42 E1 38 : 3B
30D8 03 ED 42 13 3D 20 ED EB : 7A
30E0 C9 DD E1 D1 E1 D5 E5 DD : D0
30E8 E9 DD E1 E1 D1 C1 D5 E5 : D4
30F0 C5 DD E9 DD E1 CD CA 1F : FF
30F8 6F 26 00 E5 DD E9 DD E1 : FE
SUM: 05 3E CE BB 6E F2 2C 95 3D66

3100 CD D0 1F 6F 26 00 E5 DD : 13
3108 E9 DD E1 CD 21 20 6F 26 : 4A
3110 00 E5 DD E9 DD E1 2A 06 : 99
3118 30 54 5D 19 19 7D 84 67 : 7B
3120 85 6F 11 54 00 19 22 06 : 9A
3128 30 E5 DD E9 DD E1 D1 E1 : 4B
3130 63 CD 1B 20 6F 26 00 E5 : E5
3138 DD E9 DD E1 E1 7D CD C1 : 70
3140 1F DD E9 DD E1 E1 CD RE : 0F
3148 1F DD E9 DD E1 E1 CD 73 : C4
3150 38 DD E9 DD E1 E1 7D CD : E7
3158 F4 1F DD E9 DD E1 E1 7E : F6
3160 FE 22 28 0B FE 0D CA 6F : 97
3168 31 CD F4 1F 23 18 F0 DD : 19
3170 E9 DD E1 E1 7E FE 22 28 : 4E
3178 F6 FE 0D CA 6F 31 0E 1C : 95
SUM: 53 70 C2 D1 F8 F3 A4 09 7DFA

3180 FE 52 28 16 0C FE 4C 28 : 0C
3188 11 0C FE 55 28 0C 0C FE : AE
3190 44 28 07 0E 0C FE 43 28 : F6
3198 01 0C 79 CD F4 1F 23 18 : A1
31A0 D3 3E 0D C3 F4 1F DD E1 : B2
31A8 E1 7D CD 30 20 DD E9 DD : 1E
31B0 E1 E1 45 78 B7 28 05 CD : 30
31B8 C4 1F 10 FB DD E9 DD E1 : 72
31C0 D1 E1 63 CD 1E 20 DD E9 : E6
31C8 C1 2A 02 30 2B 70 2B 71 : 54
31D0 22 02 30 EB E9 2A 02 30 : 84
31D8 5E 23 56 23 22 02 30 EB : 39
31E0 E9 D1 2A 02 30 2B 72 2B : DE
31E8 73 22 02 30 EB E9 DD E1 : 59
31F0 E1 7C B5 28 0A 2A 02 30 : A0
31F8 23 23 22 02 30 DD E9 2A : 8A
SUM: 1F 0F C3 13 85 0B DA AD 279A

3200 02 30 5E 23 56 EB E9 DD : BA
3208 E1 E1 D1 ED 73 04 30 ED : 14
3210 7B 02 30 DD E5 E5 D5 ED : 16
3218 73 02 30 ED 7B 04 30 DD : 1E
3220 E9 DD E1 ED 73 04 30 ED : 28
3228 7B 02 30 D1 E1 FD E1 13 : 50
3230 B7 ED 52 38 13 19 FD E5 : 3C
3238 E5 D5 FD 73 02 30 ED 7B : B4
3240 04 30 FD E5 DD E1 DD E9 : 9A
3248 ED 73 02 30 ED 7B 04 30 : 2E
3250 DD E9 ED 7B 00 30 3A 08 : A0
3258 30 ED 5B 09 30 2A 0B 30 : 16
3260 C9 DD E1 11 00 00 2A 02 : C4
3268 30 19 5E 23 56 D5 DD E9 : BB
3270 DD E1 11 06 00 18 EF DD : B9
3278 E1 E1 ED 73 04 30 ED 7B : BE
SUM: 86 E7 63 89 E6 F5 22 88 CE8D

3280 02 30 E5 ED 73 02 30 ED : 96
3288 7B 04 30 DD E9 DD E1 2A : 5D
3290 02 30 01 00 AE B7 ED 42 : C7
3298 ED 73 04 30 ED 7B 02 30 : 2E
32A0 E1 ED 73 02 30 ED 7B 04 : DF
32A8 30 E5 DD E9 DD E1 2A 02 : C5
32B0 30 5E 23 56 23 73 23 72 : 32
32B8 DD E9 DD E1 C1 2A 0B 30 : AA
32C0 ED 5B 09 30 3A 08 30 ED : E0
32C8 43 CC 32 CD 00 00 22 0B : 3B
32D0 30 ED 53 09 30 32 08 30 : 13
32D8 DD E9 DD E1 E1 7D 32 08 : 1C
32E0 30 DD E9 DD E1 3A 08 30 : 26
32E8 6F 26 00 E5 DD E9 DD E1 : FE
32F0 D1 ED 53 09 30 DD E9 DD : ED
32F8 E1 ED 5B 09 30 D5 DD E9 : FD
SUM: 18 CA 6C D7 51 08 0A 38 2A55

3300 DD E1 E1 22 0B 30 DD E9 : C2
3308 DD E1 2A 0B 30 E5 DD E9 : CE
3310 DD E1 E1 5E 16 00 D5 DD : C5
3318 E9 DD E1 E1 5E 23 56 D5 : 34
3320 DD E9 DD E1 E1 D1 73 DD : 86
3328 E9 DD E1 E1 D1 73 23 72 : 61
3330 DD E9 DD E1 C1 ED 58 16 : A0
3338 00 D5 DD E9 DD E1 C1 FB : PB
3340 ED 69 DD E9 DD E1 E1 6C : 27
3348 26 00 E5 DD E9 DD E1 E1 : 70
3350 26 00 E5 DD E9 DD E1 E1 : 70
3358 7D 6C 67 E5 DD E9 DD E1 : B9
3360 E1 7D 2F 6F 7C 2F 67 E5 : F3
3368 DD E9 DD E1 C1 E1 41 CB : 32
3370 3C CB 1D 10 FA E5 DD E9 : D9
3378 DD E1 C1 E1 41 29 10 FD : D7
SUM: B0 EB 3D C1 03 EC C9 4F 1A5A

3380 E5 DD E9 DD E1 CD 18 20 : 6E
3388 26 00 E5 DD E9 DD E1 CD : 5C
3390 18 20 6C 26 00 E5 DD E9 : 75
3398 DD E1 E1 CD A1 33 E5 DD : 02
33A0 E9 7D 2F 6F 7C 2F 67 23 : 39
33A8 C9 7B 2F 5F 7A 2F 57 7D : 4F
33B0 2F 6F 7C 2F 67 01 01 00 : B2
33B8 EB 09 EB 0B ED 4A C9 DD : C7
33C0 E1 E1 D1 1A 77 FE 0D 28 : 57
33C8 0A FE 22 28 04 23 13 18 : A4
33D0 F2 36 0D DD E9 DD E1 C1 : 7A
33D8 E1 D1 1A 77 FE 0D 28 0D : 83
33E0 FE 22 28 07 23 13 0B 78 : 08
33E8 B1 20 EF 36 0D DD E9 DD : A6
33F0 E1 C1 FD E1 E1 5D 54 7E : 90
33F8 FE 22 28 07 FE 0D 28 03 : 85
SUM: 18 59 36 70 26 D0 DC 14 F1D7

3400 23 18 F4 2B E5 B7 ED 52 : 35
3408 E1 28 05 0B 78 B1 20 F3 : 55
3410 FD E5 D1 EB 18 AD DD E1 : 21
3418 D9 C1 D9 C1 E1 D1 0B 78 : 69
3420 B1 28 08 1A FE 22 FE 0D : 26
3428 13 18 F3 D9 C5 D9 C1 1A : 70
3430 77 FE 22 28 0B FE 0D 28 : FD
3438 07 23 13 0B 78 B1 20 EF : 80
3440 36 0D DD E9 DD E1 D1 E1 : 79
3448 7E FE 0D CA C3 33 FE 22 : 69
3450 CA C3 33 23 18 F2 DD E1 : AB
3458 E1 01 00 00 7E FE 0D 28 : 93
3460 08 FE 22 28 04 23 03 18 : 92
3468 F3 C5 DD E9 DD E1 C1 E1 : DE
3470 11 01 00 7E B9 28 0F FE : 7E
3478 0D 28 08 FE 22 28 04 13 : 9C
SUM: 94 02 F7 6B 8E E8 71 F2 312C

3480 23 18 F0 11 00 00 D5 DD : EE
3488 E9 DD E1 E1 D1 1A FE 0D : 7E
3490 28 0C FE 22 28 08 46 90 : 5A
3498 20 13 23 13 18 EF 01 00 : 71
34A0 00 7E FE 0D 28 0F FE 22 : E0
34A8 28 0B 0B 18 FF 01 01 00 : 60
34B0 30 03 01 FF FF C5 DD E9 : BD
34B8 DD E1 C1 D1 01 09 EB C1 : E6
34C0 ED 4A E5 D5 DD E9 DD E1 : 75
34C8 D1 C1 E1 B7 ED 52 EB E1 : 35
34D0 ED 42 E5 D5 DD E9 DD E1 : 6D
34D8 D1 E1 D9 FD 21 00 00 21 : CA
34E0 00 00 D1 C1 D9 06 20 CB : 5C
34E8 3C CB 1D CB 1A CB 1B D9 : 88
34F0 30 04 FD 19 ED 4A CB 23 : 6F
34F8 CB 12 CB 11 CB 10 D9 10 : 7D
SUM: 3C 90 F7 30 94 3E 65 E1 9BED

3500 E6 D9 E5 FD E5 DD E9 DD : 29
3508 E1 CD 2C 35 D9 C5 D9 C5 : 4B
3510 DD E9 DD E1 CD 2C 35 D9 : 8B
3518 E5 D9 E5 DD E9 DD E1 CD : F4
3520 2C 35 D9 E5 D9 E5 D9 C5 : 7B
3528 D9 C5 DD E9 FD E1 D1 21 : 34
3530 00 00 D9 D1 21 00 00 D9 : A4
3538 C1 D9 C1 D9 C1 2E 30 F5 : 52
3540 21 CB 10 D9 CB 11 CB 10 : 8C
3548 D9 ED 6A D9 ED 6A D9 B7 : F0
3550 ED 52 D9 ED 52 D9 38 0E : 76
3558 03 78 B1 20 03 D9 03 D9 : 04
3560 F1 3D 20 DA FD E9 19 D9 : 00
3568 ED 5A D9 18 F3 DD E1 C1 : AA
3570 D1 E1 CD 7C 35 D9 D5 D9 : B7
3578 E5 D5 DD E9 F5 C5 D9 C1 : D4
SUM: CD 0A CA 7E D0 22 FE B4 366B

3580 21 00 00 11 00 00 D9 3E : 49
3588 20 EB 29 EB ED 6A D9 EB : 3A
3590 ED 6A EB ED 6A D5 E5 CD : 20
3598 AA 35 E1 D1 38 03 CD AA : 43
35A0 35 D9 38 01 1C 30 20 E1 : A1
35A8 F1 C9 EB B7 ED 42 EB D0 : 46
35B0 67 7D D6 01 6F 7C 26 00 : CC
35B8 C9 DD E1 C1 D1 21 00 00 : 3A
35C0 D9 11 00 00 21 00 00 D9 : E4
35C8 3E 10 CB 38 CB 19 30 05 : 6A
35D0 19 D9 ED 5A D9 CB 23 CB : CB
35D8 12 D9 CB 13 CB 12 D9 3D : BC
35E0 20 E8 D9 E5 D9 E5 DD E9 : 4A
35E8 DD E1 C1 D1 21 00 00 79 : EA
35F0 B7 28 0C CB 3F 30 01 19 : 3F
35F8 CB 23 CB 12 C3 F0 35 E5 : 98
SUM: EF 6D C3 6C 64 59 D4 97 FBF9

3600 DD E9 DD E1 C1 D1 E1 B7 : AE
3608 ED 42 44 4A E1 ED 52 38 : 18
3610 09 7C B5 B0 B1 20 08 E5 : A8
3618 DD E9 21 FF FF 18 F8 21 : 16
3620 01 00 18 F3 DD E1 E1 11 : BC
3628 00 00 CB 7C 28 01 1B D5 : 60
3630 E5 DD E9 DD E1 E1 11 00 : 5B
3638 00 7E 23 FE 0D 28 08 FE : DA
3640 22 28 04 53 5F 18 F2 D5 : DF
3648 DD E9 DD E1 D1 E1 01 01 : 38
3650 00 CB 7A 20 0D CB 7C 20 : D9
3658 06 B7 ED 52 38 01 0B C5 : 05

3660 DD E9 CB 7C 28 F8 18 F1 : 36
3668 DD E1 E1 11 01 00 7C B5 : E2
3670 28 01 1B D5 DD E9 DD E1 : 9D
3678 E1 23 E5 DD E9 DD E1 E1 : 4E
SUM: 5E 6C E3 03 A9 64 14 FC F554
3680 2B E5 DD E9 DD E1 D1 21 : 86
3688 00 00 CD A0 37 11 C4 37 : B0
3690 CD E8 1F DD E9 DD E1 D1 : 29
3698 E1 18 EF DD E1 E1 11 00 : 98
36A0 00 CB 7C 28 08 3E 2D CD : AF
36A8 F4 1F CD A1 33 EB 18 DA : 91
36B0 DD E1 D1 E1 CB 7C 28 D2 : B1
36B8 3E 2D CD F4 1F CD A9 33 : F4
36C0 C3 8A 36 DD E1 E1 D1 E5 : D8
36C8 21 00 00 CD A0 37 E1 11 : B7
36D0 C4 37 C3 C3 33 DD E1 C1 : 33
36D8 D1 E1 C5 18 EE DD E1 D1 : 0C
36E0 E1 CD BE 1F EB CD BE 1F : 20
36E8 DD E9 DD E1 FD E1 FD 7E : DD
36F0 00 FE 2D 28 06 CD 07 37 : 2D
36F8 E5 DD E9 FD 23 CD D0 37 : 9F
SUM: 04 10 0E 8B B6 3C 6C 68 C928

3700 CD A1 33 18 F3 DD E1 FD : 67
3708 E1 CD EE 37 E5 D5 DD E9 : 53
3710 DD E1 FD E1 CD 87 38 E5 : 0D
3718 DD E9 DD E1 CD 18 20 26 : AF
3720 00 ED 5B 76 1F CD D3 1F : 9C
3728 1A FE 1B 28 0C 19 EB E1 : 4C
3730 1A B7 28 06 77 13 23 18 : C4
3738 F7 E1 36 0D DD E9 DD E1 : 9F
3740 C1 D1 E1 ED B0 DD E9 DD : B3
3748 E1 C1 D1 E1 ED B8 DD E9 : BF
3750 DD E1 D1 C1 E1 ED 73 54 : 03
3758 5D 13 ED B0 DD E9 DD E1 : 91
3760 D1 E1 E5 D5 E5 D5 DD E9 : EC
3768 DD E1 E1 E1 DD E9 DD E1 : 04
3770 D1 E1 D9 D1 E1 D9 E5 D5 : D0
3778 D9 E5 D5 DD E9 CD CD 1F : 12
SUM: C7 C9 B3 65 D8 20 56 A3 84C1

3780 C0 C3 52 32 C3 D9 1F C3 : 85
3788 D6 1F DD E1 E1 CD 94 1F : 14
3790 6F 26 00 E5 DD E9 DD E1 : FE
3798 C1 E1 79 CD 9A 1F DD E9 : 67
37A0 01 00 00 C5 01 0A 00 CD : 9E
37A8 7C 35 D9 3B 30 83 D9 C1 : 15
37B0 03 F5 7C B5 B2 B3 20 EB : 99
37B8 41 21 C4 37 F1 77 23 10 : F8
37C0 FB 36 0D C9 00 00 00 00 : 07
37C8 00 00 00 00 00 00 00 00 : 00
37D0 21 00 00 FD 7E 00 FE 30 : CA
37D8 D8 FE 3A D0 FD 23 29 54 : 7D
37E0 5D 29 29 19 D6 30 85 30 : 83
37E8 01 24 6F C3 D3 37 21 00 : 82
37F0 00 11 00 00 FD 7E 00 FE : 8A
37F8 30 D8 FE 3A D0 FD 23 D6 : 06
SUM: 09 9E 9E 60 E0 6A 79 BD BD6E

3800 30 EB 29 EB ED 6A 44 4D : 17
3808 C5 42 4B EB 29 EB ED 6A : A8
3810 EB 29 EB ED 6A EB 09 EB : 35
3818 C1 ED 4A 06 00 4F EB 09 : 41
3820 EB 0E 00 ED 4A 18 CD DD : F2
3828 E5 DD 21 64 38 FD 21 6E : 0B
3830 38 3E 04 06 05 0E 2F DD : 9F
3838 5E 00 DD 56 01 0C B7 ED : 42
3840 52 30 FA 19 B7 CA 56 38 : 9E
3848 FD 71 00 DD 23 DD 23 FD : 6B
3850 23 10 E2 DD E1 C9 30 08 : E1
3858 3E 30 B9 20 04 08 0E 20 : 81
3860 C9 08 AF C9 10 27 E8 03 : 6B
3868 64 00 0A 00 01 00 00 00 : 6F
3870 00 00 00 CD 27 38 FD 21 : 4A
3878 6E 38 06 05 FD 7E 00 CD : F9
SUM: 52 8D FF 04 FC 0D A2 0E E80B

3880 F4 1F FD 23 10 F6 C9 21 : 23
3888 00 00 DD 7E 00 FE 30 D8 : 61
3890 FE 47 D0 FE 3A 38 05 FE : 88
3898 41 D8 D6 07 D6 30 29 29 : 4E
38A0 29 29 85 30 01 24 6F DD : 78
38A8 23 18 DF FD E1 CD 00 AF : 74
38B0 DD 21 BD 38 FD E5 CD 04 : A6
38B8 B0 FD E1 FD E9 06 00 00 : 7A
38C0 00 00 7F 02 C7 00 0A 00 : 52
38C8 01 02 03 04 05 06 07 07 : 23
38D0 02 02 09 07 02 01 09 07 : 27
38D8 02 00 09 0F FD E1 D1 21 : EA
38E0 ED 38 73 23 D1 73 DD 21 : FD
38E8 EC 38 18 C8 07 02 00 0F : 1C
38F0 FD E1 D1 7B FE 03 30 10 : 6B
38F8 32 10 39 DD 21 0E 39 FD : BD
SUM: 19 02 AB 67 AA A6 94 1C DFDB

3900 E5 CD 04 B0 FD E1 18 DE : 3A
3908 DD 21 CF 38 18 F1 07 02 : 17
3910 00 09 0F FD E1 21 FC 39 : 4C
3918 36 0F 06 08 2B D1 73 10 : D2
3920 FB 2B 36 0A FD E5 DD 21 : 46

3928 F3 39 CD 04 B0 FD E1 FD : 88
3930 E9 58 CB 23 16 00 19 23 : 81
3938 36 0F D1 2B 72 2B 73 10 : 61
3940 F9 18 E1 FD E1 21 F3 39 : 1D
3948 36 06 06 04 18 E3 FD E1 : 1F
3950 21 F3 39 36 00 23 36 02 : DE
3958 06 04 18 D5 FD E1 21 F3 : E9
3960 39 36 01 06 06 18 CA FD : 5B
3968 E1 21 F3 39 36 02 06 04 : 70
3970 18 BF FD E1 21 85 39 D1 : 65
3978 72 2B 73 D1 2B 72 2B 73 : 1C

SUM: FF 27 23 46 D4 EA 53 CE C9EE

3980 FD E9 FF FF FF FF FD E1 : C0
3988 21 F3 39 36 04 06 04 23 : B4
3990 C5 11 82 39 01 04 06 EB : 81
3998 ED B0 EB C1 2B 18 92 FD : 1B
39A0 E1 21 F3 39 36 03 06 06 : 73
39A8 18 E5 FD E1 21 F3 39 36 : 5E
39B0 05 06 03 18 DA FD E1 21 : FF
39B8 F3 39 36 00 23 36 01 06 : C2
39C0 02 C3 31 39 FD E1 DD E1 : CB
39C8 C3 B4 38 FD E1 21 F3 39 : DA
39D0 36 08 C1 D1 23 73 23 72 : FB
39D8 23 71 23 70 23 36 0F FD : 8C
39E0 E5 DD 21 F3 39 CD 04 B0 : 90
39E8 FD E1 3A 02 C2 6F 26 00 : 71
39F0 E5 FD E9 00 00 00 00 00 : CB
39F8 00 00 00 00 00 00 00 00 : 00

SUM: A6 8D 5F CD A2 31 E0 88 42F5

3A00 00 00 00 00 00 00 00 00 : 00
3A08 00 00 00 00 00 00 00 00 : 00
3A10 00 00 00 3E 0C CD F4 1F : ZA
3A18 CD E2 1F 2A 2A 20 53 74 : 09
3A20 61 63 6B 20 43 6F 6D 70 : DE
3A28 69 6C 65 72 20 76 65 72 : 19
3A30 20 31 2E 30 20 2A 2A 0D : 30
3A38 00 CD E2 1F 54 45 58 54 : 13
3A40 20 41 44 44 52 45 53 53 : 26
3A48 20 20 20 3A 00 CD E5 3A : 86
3A50 D8 22 91 3E CD E2 1F 4F : E6
3A58 42 4A 45 43 54 20 41 44 : 0D
3A60 44 52 45 53 53 20 3A 00 : DB
3A68 CD E5 3A 38 A6 22 93 3E : BD
3A70 CD E2 1F 56 41 52 49 41 : 41
3A78 42 4C 45 20 54 4F 50 20 : 06

SUM: 31 E1 1C 49 0E 38 99 95 E585

3A80 20 20 3A 00 CD E5 3A 38 : 9E
3A88 8A 22 95 3E CD E2 1F 53 : A0
3A90 54 41 43 4B 20 54 4F 50 : 36
3A98 20 20 20 20 20 3A 00 : FA
3AA0 CD E5 3A DA 13 3A 22 97 : CC
3AA8 3E CD E2 1F 52 45 54 5F : 56
3AB0 53 54 41 43 4B 20 54 4F : 39
3AB8 50 20 20 3A 00 CD E5 3A : B6
3AC0 DA 13 3A 22 99 3E CD E2 : CF
3AC8 1F 4F 46 46 53 45 54 20 : 06
3AD0 41 44 44 52 45 53 53 20 : 26
3AD8 3A 00 CD E5 3A DA 13 3A : 4D
3AE0 22 9B 3E 18 15 ED 5B 76 : E6
3AE8 1F CD D3 1F 1A FE 1B 37 : 48
3AF0 C8 01 10 00 EB 09 CB 33 : 7B
3AF8 B2 1F CD E2 1F 50 41 53 : 83

SUM: FB F7 2E D7 2E 9B BA 79 0EA1

3B00 53 20 31 0D 00 CD 45 3B : FE
3B08 CD E2 1F 50 41 53 53 20 : 25
3B10 32 0D 00 CD 45 3B CD E2 : 3B
3B18 1F 4F 42 4A 45 43 54 20 : F6
3B20 45 4E 44 3A 0D 00 21 52 : 91
3B28 32 FD 36 00 C3 FD 75 01 : 9B
3B30 FD 74 02 FD 23 FD 23 FD : B0
3B38 23 FD E5 E1 CD BE 1F CD : 5D
3B40 EB 1F C3 FA 1F AF 32 9D : 64
3B48 3E DD 2A 91 3E FD 2A 93 : CE
3B50 3E ED 4B 9B 3E FD 09 21 : 76
3B58 0D 30 FD 36 00 CD FD 75 : AF
3B60 01 FD 74 02 FD 23 FD 23 : B4

3B68 FD 23 FD 36 00 ED FD 36 : 73
3B70 01 73 21 00 30 FD 75 02 : 39
3B78 FD 74 03 FD 36 04 31 2A : 06

SUM: 78 3A BD 1D 89 DD 93 C5 E2A4

3B80 97 3E FD 75 05 FD 74 06 : C3
3B88 FD 36 07 21 2A 99 3E FD : 59
3B90 75 08 FD 74 09 FD 36 0A : 34
3B98 22 21 02 30 FD 75 0B FD : EF
3BA0 74 0C 01 0D 00 FD 09 DD : 71
3BA8 7E 00 DD 23 B7 C8 FE FF : FA
3BB0 20 1E 3A 9D 3E B7 28 EF : 21
3BB8 FD E5 E1 ED 4B 9B 3E B7 : 8B
3BC0 ED 42 EB E1 73 23 72 3A : 3D
3BC8 9D 3E 3D 32 9D 3E 18 E2 : 1F
3BD0 FE 88 CA 43 3C FE 89 CA : 20
3BD8 E5 3C FE 8A CA 8A 3C FE : 37
3BE0 8B CA 65 3C FE 8C CA E7 : 31
3BE8 3D FE 8D CA 6A 3D FE 8E : C5
3BF0 CA 37 3D FE 8F CA 0A 3D : DC
3BF8 FE 90 CA A1 3D FE 91 CA : 8F

SUM: 37 7F E5 79 BF 99 12 EC 7013

3C00 C4 3D FE 92 CA 82 3D FE : 18
3C08 93 CA 50 3D FE 94 CA 10 : 56
3C10 3D FE 23 CA 1A 3E FE 10 : 8E
3C18 CA 30 3E FE 0E CA 45 3E : 91
3C20 FE 01 CA 4E 3E 6F 26 00 : EA
3C28 29 01 9E 3E 09 FD 36 00 : 42
3C30 CD 7E FD 77 01 23 7E FD : 5E
3C38 77 02 FD 23 FD 23 FD 23 : D9
3C40 C3 A7 3B FD E5 E1 ED 4B : A0
3C48 9B 3E B7 ED 42 EB DD 6E : F5
3C50 00 DD 66 01 29 DD 23 DD : 4A
3C58 23 7B CD 9A 1F 23 7A CD : 8E
3C60 9A 1F C3 A7 3B FD 36 00 : 91
3C68 C3 DD 6E 00 DD 66 01 29 : 7B
3C70 DD 23 DD 23 CD 94 1F FD : 7D
3C78 77 01 23 CD 94 1F FD 77 : 8F

SUM: FB 14 67 D9 1D B2 DB 7C CE40

3C80 02 FD 23 FD 23 FD 23 C3 : 25
3C88 A7 3B FD 36 00 21 FD 23 : 56
3C90 FD E5 E1 ED 4B 9B 3E B7 : 8B
3C98 ED 42 01 06 00 09 FD 75 : B1
3CA0 00 FD 74 01 FD 36 02 5E : 8C
3CA8 FD 23 FD 23 FD 23 FD 36 : 93
3CB0 00 C3 FD 23 FD E5 FD 23 : E5
3CB8 FD 23 DD 7E 00 FE 22 28 : C3
3CC0 09 FD 77 00 DD 23 FD 23 : 9D
3CC8 18 F0 FD 36 00 0D DD 23 : 48
3CD0 FD 23 FD E5 E1 ED 4B 9B : B6
3CD8 3E B7 B7 ED 42 D1 EB 73 : 0A
3CE0 23 72 C3 A7 3B FD 36 00 : 6D
3CE8 E1 FD 36 01 7C FD 36 02 : C6
3CF0 B5 FD 36 03 CA 01 04 00 : BA
3CF8 FD 09 FD E5 FD 23 FD 23 : 28

SUM: 9F A1 A1 83 E3 0A F6 F1 E06D

3D00 3A 9D 3E 3C 32 9D 3E C3 : 21
3D08 A7 3B CD 19 3D C3 A7 3B : AA
3D10 CD 19 3D CD 19 3D C3 A7 : B0
3D18 3B FD 36 00 21 DD 7E 00 : EA
3D20 FD 77 01 DD 7E 01 FD 77 : 45
3D28 02 FD 36 03 E5 DD 23 DD : FA
3D30 23 01 04 00 FD 09 C9 CD : C4
3D38 65 3E FD 36 00 2A FD 75 : 72
3D40 01 FD 74 02 FD 36 03 E5 : 8F
3D48 01 04 00 FD 09 C3 A7 3B : B0
3D50 CD 65 3E FD 36 00 2A FD : CA
3D58 75 01 FD 74 02 FD 36 03 : 1F
3D60 E5 23 23 01 04 00 FD 09 : 36
3D68 18 0D CD 65 3E CD 75 3E : D8
3D70 FD 36 00 22 FD 75 01 FD : C5
3D78 74 02 01 03 00 FD 09 C3 : 43

SUM: 22 33 56 33 86 C0 92 62 83D1

3D80 A7 3B CD 65 3E 23 23 CD : 65

3D88 75 3E FD 36 00 22 FD 75 : 7A
3D90 01 FD 74 02 FD 36 03 E1 : 8B
3D98 2B 2B 01 04 00 FD 09 18 : 79
3DA0 CF CD 65 3E FD 36 00 2A : 9C
3DA8 FD 75 01 FD 74 02 FD 36 : 19
3DB0 03 23 FD 36 04 22 FD 75 : F1
3DB8 05 FD 74 06 01 07 00 FD : 81
3DC0 09 C3 A7 3B CD 65 3E FD : 1B
3DC8 36 00 2A FD 75 01 FD 74 : 4A
3DD0 02 FD 36 03 2B FD 36 04 : 94
3DD8 22 FD 75 05 FD 74 06 01 : 11
3DE0 07 00 FD 09 C3 A7 3B DD : 8F
3DE8 6E 00 DD 66 01 29 CD 94 : 3C
3DF0 1F 5F 23 CD 94 1F 57 FD : 75
3DF8 36 00 11 FD 73 01 FD 72 : 27

SUM: 49 1F A0 91 E6 A0 F9 63 7FD3

3E00 02 FD 36 03 CD 21 C8 31 : 1F
3E08 FD 75 04 FD 74 05 DD 23 : EC
3E10 DD 23 01 06 00 FD 09 C3 : D0
3E18 A7 3B 21 D5 31 FD 36 00 : 3C
3E20 C3 FD 75 01 FD 74 02 FD : A6
3E28 23 FD 23 FD 23 C3 A7 3B : 08
3E30 FD 36 00 E1 FD 36 01 E5 : 2D
3E38 FD 36 02 E5 FD 23 FD 23 : 5A
3E40 FD 23 C3 A7 3B FD 36 00 : F8
3E48 E1 FD 23 C3 A7 3B CD 75 : E8
3E50 3E FD 36 00 D1 FD 36 01 : F6
3E58 19 FD 36 02 E5 01 03 00 : 37
3E60 FD 09 C3 A7 3B DD 6E 00 : F7
3E68 DD 66 01 ED 4B 95 3E 09 : 58
3E70 DD 23 DD 23 C9 FD 7E FF : 43
3E78 FE E5 20 0E FD 7E FC FE : 86

SUM: 4D C7 09 D0 70 D3 ED D3 7039

3E80 C3 28 07 FE CD 28 03 FD : E5
3E88 2B C9 FD 36 00 E1 FD 23 : 28
3E90 C9 00 00 00 00 00 00 00 : C9
3E98 00 00 00 00 00 00 00 00 : 00
3EA0 00 00 20 30 2A 30 34 30 : 0E
3EA8 3E 30 48 30 53 30 63 30 : FC
3EB0 73 30 79 30 89 30 96 30 : CB
3EB8 A3 30 00 00 E1 30 00 00 : E4
3EC0 F3 30 FE 30 09 31 14 31 : 56
3EC8 2C 31 3A 31 43 31 4B 31 : B8
3ED0 53 31 5C 31 71 31 00 00 : B3
3ED8 00 00 A6 31 AF 31 BE 31 : A6
3EE0 00 00 C8 31 D5 31 00 00 : FF
3EE8 E1 31 EE 31 BA 32 DA 32 : 29
3EF0 E3 32 EE 32 F7 32 00 33 : 91
3EF8 08 33 10 33 19 33 22 33 : 1F

SUM: 49 A9 D3 4E BF 55 46 DB 8854

3F00 29 33 32 33 3C 33 07 32 : 69
3F08 21 32 44 33 4D 33 55 33 : D2
3F10 5E 33 6A 33 78 33 83 33 : 8F
3F18 8D 33 98 33 BF 33 D5 33 : 85
3F20 EF 33 44 34 56 34 6C 34 : C4
3F28 89 34 B8 34 C6 34 D6 34 : AD
3F30 07 35 12 35 B9 35 24 36 : CB
3F38 33 36 4A 36 68 36 84 36 : 41
3F40 95 36 9B 36 B0 36 C3 36 : 7B
3F48 D5 36 DD 36 EA 36 05 37 : 7A
3F50 1A 37 3E 37 47 37 50 37 : C8
3F58 5E 37 68 37 6E 37 AB 38 : BC
3F60 DC 38 F0 38 13 39 43 39 : 04
3F68 4E 39 5C 39 67 39 72 39 : 67
3F70 9F 39 86 39 AA 39 CB 39 : 7E
3F78 B5 39 C4 39 84 37 87 37 : 64

SUM: 47 5A 84 5C F4 5B 68 5D 0C73

3F80 8A 37 96 37 52 32 61 32 : A5
3F88 70 32 A1 31 E9 30 77 32 : 36
3F90 8D 32 AC 32 02 36 1D 35 : 27
3F98 6D 35 E8 35 76 36 7D 36 : 1E
3FA0 40 37 7D 37 16 34 00 00 : 45
3FA8 00 00 00 00 00 00 00 : 00

SUM: 04 07 48 06 C9 02 72 CF 874F

リスト2 ソースリスト1

```
3000 1 ORG #3000
3000 2 OFFSET #0000
3000 3
3000 4 #PRINT EQU $1F4
3000 5 #PRINT EQU $1F2
3000 6 #MSG EQU $1F8
3000 7 #NL EQU $1F8
3000 8 #CTRL EQU $1D2
3000 9 #PRTHL EQU $1F8
3000 10 #HLTHX EQU $1F2
3000 11 #KBPAD EQU $1F76
3000 12 #PAUSE EQU $1FCT
3000 13 #RELL EQU $1FCA
3000 14 #HOT EQU $1FPA
3000 15 #FILE EQU $1FA3
3000 16 #KOPEN EQU $1FAF
3000 17 #MSG EQU $1FAC
3000 18 #OPEN EQU $2009
3000 19 #RD EQU $1FAF
3000 20 #PRINT EQU $1F9D
3000 21 #OTADR EQU $1F70
3000 22 #SIZE EQU $1F72
3000 23 #XADR EQU $1F6E
3000 24 #LPSW EQU $1F7C
3000 25 #MSX EQU $1F85
3000 26 #PRTHX EQU $1FC1
3000 27 #BREAK EQU $1FCD
```

```
3000 28 #GETRY EQU $1FDD
3000 29 #INKEY EQU $1FCA
3000 30 #FGET EQU $2021
3000 31 #HEX EQU $1F85
3000 32 #CSR EQU $2016
3000 33 #PONE EQU $1F9A
3000 34 #PEER EQU $1F94
3000 35 #WIDCH EQU $2010
3000 36 #LOC EQU $201E
3000 37 #WIDTH EQU $1F5C
3000 38 #SCRN EQU $201B
3000 39 #DIR EQU $2006
3000 40 #RET SP EQU $A200
3000 41 #VAR SP EQU $A200
3000 42
3000 43 ; STACK Compiler Run Time Routine
3000 44
3000 45 MAIN SP DW 0
3000 46 RET SP DW $A200
3000 47 STX WR DW 0
3000 48 RNDR DW 0
3000 49 #A DB 0
3000 50 #DE DW 0
3000 51 #HL DW 0
3000 52
3000 53 #START
```

```
3000 32 08 30 54 LD (#A),A
3000 38 53 09 55 LD (#DE),DE
3000 38
3000 14 22 0B 30 56 LD (#HL),HL
3000 37 C9 57 RET
3000 38 58 ;
3000 38 59 ; エンジン (1)
3000 38 60 ;
3000 38 61 #TASU
3000 38 DD E1 62 POP IX
3000 38 63 POP DE
3000 38 E1 64 POP HL
3000 38 19 65 ADD HL,DE
3000 38 DD E9 66 PUSH HL
3000 38 DD E9 67 JP (IX)
3000 38 68 #HIKU
3000 38 DD E1 69 POP IX
3000 38 D1 70 POP DE
3000 38 E1 71 POP HL
3000 38 72 72 OR A
3000 38 DD 52 73 SBC HL,DE
3000 38 E5 74 PUSH HL
3000 38 DD E9 75 JP (IX)
3000 38 76 #MLT
3000 38 DD E1 77 POP IX
3000 38 D1 78 POP DE
3000 38 E1 79 POP HL
```



```

302E CD B0 30 80 CALL MLT
3031 E5 81 PUSH HL
3032 DD E9 82 JP (IX)
3033 83 83 DIV
3034 DD E1 84 POP IX
3035 D1 85 POP DE
3036 E1 86 POP HL
3037 CD C3 30 87 CALL DIV
3038 E5 88 PUSH HL
3039 DD E9 89 JP (IX)
303A 90 90 MOD
303B DD E1 91 POP IX
303C D1 92 POP DE
303D E1 93 POP HL
303E CD C3 30 94 CALL DIV
303F D5 95 PUSH HL
3040 DD E9 96 JP (IX)
3041 98 98 DIVMOD
3042 DD E1 99 POP IX
3043 D1 100 POP DE
3044 E1 101 POP HL
3045 CD C3 30 102 CALL DIV
3046 E5 103 PUSH HL
3047 D5 104 PUSH DE
3048 DD E9 105 JP (IX)
3049 106 106 MOD
3050 DD E1 107 POP IX
3051 E1 108 POP DE
3052 D1 109 POP HL
3053 CD C3 30 110 CALL DIV
3054 E5 111 PUSH HL
3055 D5 112 PUSH DE
3056 DD E9 113 JP (IX)
3057 114 114 MOD
3058 DD E1 115 POP IX
3059 E1 116 POP DE
3060 D1 117 POP HL
3061 CD C3 30 118 CALL DIV
3062 E5 119 PUSH HL
3063 D5 120 PUSH DE
3064 DD E9 121 JP (IX)
3065 122 122 MOD
3066 DD E1 123 POP IX
3067 E1 124 POP DE
3068 D1 125 POP HL
3069 CD C3 30 126 CALL DIV
3070 E5 127 PUSH HL
3071 D5 128 PUSH DE
3072 DD E9 129 JP (IX)
3073 130 130 MOD
3074 DD E1 131 POP IX
3075 E1 132 POP DE
3076 D1 133 POP HL
3077 CD C3 30 134 CALL DIV
3078 E5 135 PUSH HL
3079 D5 136 PUSH DE
3080 DD E9 137 JP (IX)
3081 138 138 MOD
3082 DD E1 139 POP IX
3083 E1 140 POP DE
3084 D1 141 POP HL
3085 CD C3 30 142 CALL DIV
3086 E5 143 PUSH HL
3087 D5 144 PUSH DE
3088 DD E9 145 JP (IX)
3089 146 146 MOD
3090 DD E1 147 POP IX
3091 E1 148 POP DE
3092 D1 149 POP HL
3093 CD C3 30 150 CALL DIV
3094 E5 151 PUSH HL
3095 D5 152 PUSH DE
3096 DD E9 153 JP (IX)
3097 154 154 MOD
3098 DD E1 155 POP IX
3099 E1 156 POP DE
3100 D1 157 POP HL
3101 CD C3 30 158 CALL DIV
3102 E5 159 PUSH HL
3103 D5 160 PUSH DE
3104 DD E9 161 JP (IX)
3105 162 162 MOD
3106 DD E1 163 POP IX
3107 E1 164 POP DE
3108 D1 165 POP HL
3109 CD C3 30 166 CALL DIV
3110 E5 167 PUSH HL
3111 D5 168 PUSH DE
3112 DD E9 169 JP (IX)
3113 170 170 MOD
3114 DD E1 171 POP IX
3115 E1 172 POP DE
3116 D1 173 POP HL
3117 CD C3 30 174 CALL DIV
3118 E5 175 PUSH HL
3119 D5 176 PUSH DE
3120 DD E9 177 JP (IX)
3121 178 178 MOD
3122 DD E1 179 POP IX
3123 E1 180 POP DE
3124 D1 181 POP HL
3125 CD C3 30 182 CALL DIV
3126 E5 183 PUSH HL
3127 D5 184 PUSH DE
3128 DD E9 185 JP (IX)
3129 186 186 MOD
3130 DD E1 187 POP IX
3131 E1 188 POP DE
3132 D1 189 POP HL
3133 CD C3 30 190 CALL DIV
3134 E5 191 PUSH HL
3135 D5 192 PUSH DE
3136 DD E9 193 JP (IX)
3137 194 194 MOD
3138 DD E1 195 POP IX
3139 E1 196 POP DE
3140 D1 197 POP HL
3141 CD C3 30 198 CALL DIV
3142 E5 199 PUSH HL
3143 D5 200 PUSH DE
3144 DD E9 201 JP (IX)
3145 202 202 MOD
3146 DD E1 203 POP IX
3147 E1 204 POP DE
3148 D1 205 POP HL
3149 CD C3 30 206 CALL DIV
3150 E5 207 PUSH HL
3151 D5 208 PUSH DE
3152 DD E9 209 JP (IX)
3153 210 210 MOD
3154 DD E1 211 POP IX
3155 E1 212 POP DE
3156 D1 213 POP HL
3157 CD C3 30 214 CALL DIV
3158 E5 215 PUSH HL
3159 D5 216 PUSH DE
3160 DD E9 217 JP (IX)
3161 218 218 MOD
3162 DD E1 219 POP IX
3163 E1 220 POP DE
3164 D1 221 POP HL
3165 CD C3 30 222 CALL DIV
3166 E5 223 PUSH HL
3167 D5 224 PUSH DE
3168 DD E9 225 JP (IX)
3169 226 226 MOD
3170 DD E1 227 POP IX
3171 E1 228 POP DE
3172 D1 229 POP HL
3173 CD C3 30 230 CALL DIV
3174 E5 231 PUSH HL
3175 D5 232 PUSH DE
3176 DD E9 233 JP (IX)
3177 234 234 MOD
3178 DD E1 235 POP IX
3179 E1 236 POP DE
3180 D1 237 POP HL
3181 CD C3 30 238 CALL DIV
3182 E5 239 PUSH HL
3183 D5 240 PUSH DE
3184 DD E9 241 JP (IX)
3185 242 242 MOD
3186 DD E1 243 POP IX
3187 E1 244 POP DE
3188 D1 245 POP HL
3189 CD C3 30 246 CALL DIV
3190 E5 247 PUSH HL
3191 D5 248 PUSH DE
3192 DD E9 249 JP (IX)
3193 250 250 MOD
3194 DD E1 251 POP IX
3195 E1 252 POP DE
3196 D1 253 POP HL
3197 CD C3 30 254 CALL DIV
3198 E5 255 PUSH HL
3199 D5 256 PUSH DE
3200 DD E9 257 JP (IX)
3201 258 258 MOD
3202 DD E1 259 POP IX
3203 E1 260 POP DE
3204 D1 261 POP HL
3205 CD C3 30 262 CALL DIV
3206 E5 263 PUSH HL
3207 D5 264 PUSH DE
3208 DD E9 265 JP (IX)
3209 266 266 MOD
3210 DD E1 267 POP IX
3211 E1 268 POP DE
3212 D1 269 POP HL
3213 CD C3 30 270 CALL DIV
3214 E5 271 PUSH HL
3215 D5 272 PUSH DE
3216 DD E9 273 JP (IX)
3217 274 274 MOD
3218 DD E1 275 POP IX
3219 E1 276 POP DE
3220 D1 277 POP HL
3221 CD C3 30 278 CALL DIV
3222 E5 279 PUSH HL
3223 D5 280 PUSH DE
3224 DD E9 281 JP (IX)
3225 282 282 MOD
3226 DD E1 283 POP IX
3227 E1 284 POP DE
3228 D1 285 POP HL
3229 CD C3 30 286 CALL DIV
3230 E5 287 PUSH HL
3231 D5 288 PUSH DE
3232 DD E9 289 JP (IX)
3233 290 290 MOD
3234 DD E1 291 POP IX
3235 E1 292 POP DE
3236 D1 293 POP HL
3237 CD C3 30 294 CALL DIV
3238 E5 295 PUSH HL
3239 D5 296 PUSH DE
3240 DD E9 297 JP (IX)
3241 298 298 MOD
3242 DD E1 299 POP IX
3243 E1 300 POP DE
3244 D1 301 POP HL
3245 CD C3 30 302 CALL DIV
3246 E5 303 PUSH HL
3247 D5 304 PUSH DE
3248 DD E9 305 JP (IX)
3249 306 306 MOD
3250 DD E1 307 POP IX
3251 E1 308 POP DE
3252 D1 309 POP HL
3253 CD C3 30 310 CALL DIV
3254 E5 311 PUSH HL
3255 D5 312 PUSH DE
3256 DD E9 313 JP (IX)
3257 314 314 MOD
3258 DD E1 315 POP IX
3259 E1 316 POP DE
3260 D1 317 POP HL
3261 CD C3 30 318 CALL DIV
3262 E5 319 PUSH HL
3263 D5 320 PUSH DE
3264 DD E9 321 JP (IX)
3265 322 322 MOD
3266 DD E1 323 POP IX
3267 E1 324 POP DE
3268 D1 325 POP HL
3269 CD C3 30 326 CALL DIV
3270 E5 327 PUSH HL
3271 D5 328 PUSH DE
3272 DD E9 329 JP (IX)
3273 330 330 MOD
3274 DD E1 331 POP IX
3275 E1 332 POP DE
3276 D1 333 POP HL
3277 CD C3 30 334 CALL DIV
3278 E5 335 PUSH HL
3279 D5 336 PUSH DE
3280 DD E9 337 JP (IX)
3281 338 338 MOD
3282 DD E1 339 POP IX
3283 E1 340 POP DE
3284 D1 341 POP HL
3285 CD C3 30 342 CALL DIV
3286 E5 343 PUSH HL
3287 D5 344 PUSH DE
3288 DD E9 345 JP (IX)
3289 346 346 MOD
3290 DD E1 347 POP IX
3291 E1 348 POP DE
3292 D1 349 POP HL
3293 CD C3 30 350 CALL DIV
3294 E5 351 PUSH HL
3295 D5 352 PUSH DE
3296 DD E9 353 JP (IX)
3297 354 354 MOD
3298 DD E1 355 POP IX
3299 E1 356 POP DE
3300 D1 357 POP HL
3301 CD C3 30 358 CALL DIV
3302 E5 359 PUSH HL
3303 D5 360 PUSH DE
3304 DD E9 361 JP (IX)
3305 362 362 MOD
3306 DD E1 363 POP IX
3307 E1 364 POP DE
3308 D1 365 POP HL
3309 CD C3 30 366 CALL DIV
3310 E5 367 PUSH HL
3311 D5 368 PUSH DE
3312 DD E9 369 JP (IX)
3313 370 370 MOD
3314 DD E1 371 POP IX
3315 E1 372 POP DE
3316 D1 373 POP HL
3317 CD C3 30 374 CALL DIV
3318 E5 375 PUSH HL
3319 D5 376 PUSH DE
3320 DD E9 377 JP (IX)
3321 378 378 MOD
3322 DD E1 379 POP IX
3323 E1 380 POP DE
3324 D1 381 POP HL
3325 CD C3 30 382 CALL DIV
3326 E5 383 PUSH HL
3327 D5 384 PUSH DE
3328 DD E9 385 JP (IX)
3329 386 386 MOD
3330 DD E1 387 POP IX
3331 E1 388 POP DE
3332 D1 389 POP HL
3333 CD C3 30 390 CALL DIV
3334 E5 391 PUSH HL
3335 D5 392 PUSH DE
3336 DD E9 393 JP (IX)
3337 394 394 MOD
3338 DD E1 395 POP IX
3339 E1 396 POP DE
3340 D1 397 POP HL
3341 CD C3 30 398 CALL DIV
3342 E5 399 PUSH HL
3343 D5 400 PUSH DE
3344 DD E9 401 JP (IX)
3345 402 402 MOD
3346 DD E1 403 POP IX
3347 E1 404 POP DE
3348 D1 405 POP HL
3349 CD C3 30 406 CALL DIV
3350 E5 407 PUSH HL
3351 D5 408 PUSH DE
3352 DD E9 409 JP (IX)
3353 410 410 MOD
3354 DD E1 411 POP IX
3355 E1 412 POP DE
3356 D1 413 POP HL
3357 CD C3 30 414 CALL DIV
3358 E5 415 PUSH HL
3359 D5 416 PUSH DE
3360 DD E9 417 JP (IX)
3361 418 418 MOD
3362 DD E1 419 POP IX
3363 E1 420 POP DE
3364 D1 421 POP HL
3365 CD C3 30 422 CALL DIV
3366 E5 423 PUSH HL
3367 D5 424 PUSH DE
3368 DD E9 425 JP (IX)
3369 426 426 MOD
3370 DD E1 427 POP IX
3371 E1 428 POP DE
3372 D1 429 POP HL
3373 CD C3 30 430 CALL DIV
3374 E5 431 PUSH HL
3375 D5 432 PUSH DE
3376 DD E9 433 JP (IX)
3377 434 434 MOD
3378 DD E1 435 POP IX
3379 E1 436 POP DE
3380 D1 437 POP HL
3381 CD C3 30 438 CALL DIV
3382 E5 439 PUSH HL
3383 D5 440 PUSH DE
3384 DD E9 441 JP (IX)
3385 442 442 MOD
3386 DD E1 443 POP IX
3387 E1 444 POP DE
3388 D1 445 POP HL
3389 CD C3 30 446 CALL DIV
3390 E5 447 PUSH HL
3391 D5 448 PUSH DE
3392 DD E9 449 JP (IX)
3393 450 450 MOD
3394 DD E1 451 POP IX
3395 E1 452 POP DE
3396 D1 453 POP HL
3397 CD C3 30 454 CALL DIV
3398 E5 455 PUSH HL
3399 D5 456 PUSH DE
3400 DD E9 457 JP (IX)
3401 458 458 MOD
3402 DD E1 459 POP IX
3403 E1 460 POP DE
3404 D1 461 POP HL
3405 CD C3 30 462 CALL DIV
3406 E5 463 PUSH HL
3407 D5 464 PUSH DE
3408 DD E9 465 JP (IX)
3409 466 466 MOD
3410 DD E1 467 POP IX
3411 E1 468 POP DE
3412 D1 469 POP HL
3413 CD C3 30 470 CALL DIV
3414 E5 471 PUSH HL
3415 D5 472 PUSH DE
3416 DD E9 473 JP (IX)
3417 474 474 MOD
3418 DD E1 475 POP IX
3419 E1 476 POP DE
3420 D1 477 POP HL
3421 CD C3 30 478 CALL DIV
3422 E5 479 PUSH HL
3423 D5 480 PUSH DE
3424 DD E9 481 JP (IX)
3425 482 482 MOD
3426 DD E1 483 POP IX
3427 E1 484 POP DE
3428 D1 485 POP HL
3429 CD C3 30 486 CALL DIV
3430 E5 487 PUSH HL
3431 D5 488 PUSH DE
3432 DD E9 489 JP (IX)
3433 490 490 MOD
3434 DD E1 491 POP IX
3435 E1 492 POP DE
3436 D1 493 POP HL
3437 CD C3 30 494 CALL DIV
3438 E5 495 PUSH HL
3439 D5 496 PUSH DE
3440 DD E9 497 JP (IX)
3441 498 498 MOD
3442 DD E1 499 POP IX
3443 E1 500 POP DE
3444 D1 501 POP HL
3445 CD C3 30 502 CALL DIV
3446 E5 503 PUSH HL
3447 D5 504 PUSH DE
3448 DD E9 505 JP (IX)
3449 506 506 MOD
3450 DD E1 507 POP IX
3451 E1 508 POP DE
3452 D1 509 POP HL
3453 CD C3 30 510 CALL DIV
3454 E5 511 PUSH HL
3455 D5 512 PUSH DE
3456 DD E9 513 JP (IX)
3457 514 514 MOD
3458 DD E1 515 POP IX
3459 E1 516 POP DE
3460 D1 517 POP HL
3461 CD C3 30 518 CALL DIV
3462 E5 519 PUSH HL
3463 D5 520 PUSH DE
3464 DD E9 521 JP (IX)
3465 522 522 MOD
3466 DD E1 523 POP IX
3467 E1 524 POP DE
3468 D1 525 POP HL
3469 CD C3 30 526 CALL DIV
3470 E5 527 PUSH HL
3471 D5 528 PUSH DE
3472 DD E9 529 JP (IX)
3473 530 530 MOD
3474 DD E1 531 POP IX
3475 E1 532 POP DE
3476 D1 533 POP HL
3477 CD C3 30 534 CALL DIV
3478 E5 535 PUSH HL
3479 D5 536 PUSH DE
3480 DD E9 537 JP (IX)
3481 538 538 MOD
3482 DD E1 539 POP IX
3483 E1 540 POP DE
3484 D1 541 POP HL
3485 CD C3 30 542 CALL DIV
3486 E5 543 PUSH HL
3487 D5 544 PUSH DE
3488 DD E9 545 JP (IX)
3489 546 546 MOD
3490 DD E1 547 POP IX
3491 E1 548 POP DE
3492 D1 549 POP HL
3493 CD C3 30 550 CALL DIV
3494 E5 551 PUSH HL
3495 D5 552 PUSH DE
3496 DD E9 553 JP (IX)
3497 554 554 MOD
3498 DD E1 555 POP IX
3499 E1 556 POP DE
3500 D1 557 POP HL
3501 CD C3 30 558 CALL DIV
3502 E5 559 PUSH HL
3503 D5 560 PUSH DE
3504 DD E9 561 JP (IX)
3505 562 562 MOD
3506 DD E1 563 POP IX
3507 E1 564 POP DE
3508 D1 565 POP HL
3509 CD C3 30 566 CALL DIV
3510 E5 567 PUSH HL
3511 D5 568 PUSH DE
3512 DD E9 569 JP (IX)
3513 570 570 MOD
3514 DD E1 571 POP IX
3515 E1 572 POP DE
3516 D1 573 POP HL
3517 CD C3 30 574 CALL DIV
3518 E5 575 PUSH HL
3519 D5 576 PUSH DE
3520 DD E9 577 JP (IX)
3521 578 578 MOD
3522 DD E1 579 POP IX
3523 E1 580 POP DE
3524 D1 581 POP HL
3525 CD C3 30 582 CALL DIV
3526 E5 583 PUSH HL
3527 D5 584 PUSH DE
3528 DD E9 585 JP (IX)
3529 586 586 MOD
3530 DD E1 587 POP IX
3531 E1 588 POP DE
3532 D1 589 POP HL
3533 CD C3 30 590 CALL DIV
3534 E5 591 PUSH HL
3535 D5 592 PUSH DE
3536 DD E9 593 JP (IX)
3537 594 594 MOD
3538 DD E1 595 POP IX
3539 E1 596 POP DE
3540 D1 597 POP HL
3541 CD C3 30 598 CALL DIV
3542 E5 599 PUSH HL
3543 D5 600 PUSH DE
3544 DD E9 601 JP (IX)
3545 602 602 MOD
3546 DD E1 603 POP IX
3547 E1 604 POP DE
3548 D1 605 POP HL
3549 CD C3 30 606 CALL DIV
3550 E5 607 PUSH HL
3551 D5 608 PUSH DE
3552 DD E9 609 JP (IX)
3553 610 610 MOD
3554 DD E1 611 POP IX
3555 E1 612 POP DE
3556 D1 613 POP HL
3557 CD C3 30 614 CALL DIV
3558 E5 615 PUSH HL
3559 D5 616 PUSH DE
3560 DD E9 617 JP (IX)
3561 618 618 MOD
3562 DD E1 619 POP IX
3563 E1 620 POP DE
3564 D1 621 POP HL
3565 CD C3 30 622 CALL DIV
3566 E5 623 PUSH HL
3567 D5 624 PUSH DE
3568 DD E9 625 JP (IX)
3569 626 626 MOD
3570 DD E1 627 POP IX
3571 E1 628 POP DE
3572 D1 629 POP HL
3573 CD C3 30 628 CALL DIV
3574 E5 629 PUSH HL
3575 D5 630 PUSH DE
3576 DD E9 629 JP (IX)
3577 630 630 MOD
3578 DD E1 631 POP IX
3579 E1 632 POP DE
3580 D1 633 POP HL
3581 CD C3 30 634 CALL DIV
3582 E5 635 PUSH HL
3583 D5 636 PUSH DE
3584 DD E9 635 JP (IX)
3585 636 636 MOD
3586 DD E1 637 POP IX
3587 E1 638 POP DE
3588 D1 639 POP HL
3589 CD C3 30 638 CALL DIV
3590 E5 639 PUSH HL
3591 D5 640 PUSH DE
3592 DD E9 639 JP (IX)
3593 640 640 MOD
3594 DD E1 641 POP IX
3595 E1 642 POP DE
3596 D1 643 POP HL
3597 CD C3 30 644 CALL DIV
3598 E5 645 PUSH HL
3599 D5 646 PUSH DE
3600 DD E9 645 JP (IX)
3601 646 646 MOD
3602 DD E1 647 POP IX
3603 E1 648 POP DE
3604 D1 649 POP HL
3605 CD C3 30 646 CALL DIV
3606 E5 647 PUSH HL
3607 D5 648 PUSH DE
3608 DD E9 647 JP (IX)
3609 648 648 MOD
3610 DD E1 649 POP IX
3611 E1 650 POP DE
3612 D1 651 POP HL
3613 CD C3 30 648 CALL DIV
3614 E5 649 PUSH HL
3615 D5 650 PUSH DE
3616 DD E9 649 JP (IX)
3617 650 650 MOD
3618 DD E1 651 POP IX
3619 E1 652 POP DE
3620 D1 653 POP HL
3621 CD C3 30 648 CALL DIV
3622 E5 651 PUSH HL
3623 D5 652 PUSH DE
3624 DD E9 651 JP (IX)
3625 652 652 MOD
3626 DD E1 653 POP IX
3627 E1 654 POP DE
3628 D1 655 POP HL
3629 CD C3 30 648 CALL DIV
3630 E5 653 PUSH HL
3631 D5 654 PUSH DE
3632 DD E9 653 JP (IX)
3633 654 654 MOD
3634 DD E1 655 POP IX
3635 E1 656 POP DE
3636 D1 657 POP HL
3637 CD C3 30 648 CALL DIV
3638 E5 655 PUSH HL
3639 D5 656 PUSH DE
3640 DD E9 655 JP (IX)
3641 656 656 MOD
3642 DD E1 657 POP IX
3643 E1 658 POP DE
3644 D1 659 POP HL
3645 CD C3 30 648 CALL DIV
3646 E5 657 PUSH HL
3647 D5 658 PUSH DE
3648 DD E9 657 JP (IX)
3649 658 658 MOD
3650 DD E1 659 POP IX
3651 E1 660 POP DE
3652 D1 661 POP HL
3653 CD C3 30 648 CALL DIV
3654 E5 659 PUSH HL
3655 D5 660 PUSH DE
3656 DD E9 659 JP (IX)
3657 660 660 MOD
3658 DD E1 661 POP IX
3659 E1 662 POP DE
3660 D1 663 POP HL
3661 CD C3 30 648 CALL DIV
3662 E5 661 PUSH HL
3663 D5 662 PUSH DE
3664 DD E9 661 JP (IX)
3665 662 662 MOD
3666 DD E1 663 POP IX
3667 E1 664 POP DE
3668 D1 665 POP HL
3669 CD C3 30 648 CALL DIV
3670 E5 663 PUSH HL
3671 D5 664 PUSH DE
3672 DD E9 663 JP (IX)
3673 664 664 MOD
3674 DD E1 665 POP IX
3675 E1 666 POP DE
3676 D1 667 POP HL
3677 CD C3 30 648 CALL DIV
3678 E5 665 PUSH HL
3679 D5 666 PUSH DE
3680 DD E9 665 JP (IX)
3681 666 666 MOD
3682 DD E1 667 POP IX
3683 E1 668 POP DE
3684 D1 669 POP HL
3685 CD C3 30 648 CALL DIV
3686 E5 667 PUSH HL
3687 D5 668 PUSH DE
3688 DD E9 667 JP (IX)
3689 668 668 MOD
3690 DD E1 669 POP IX
3691 E1 670 POP DE
3692 D1 671 POP HL
3693 CD C3 30 648 CALL DIV
3694 E5 669 PUSH HL
3695 D5 670 PUSH DE
3696 DD E9 669 JP (IX)
3697 670 670 MOD
3698 DD E1 671 POP IX
3699 E1 672 POP DE
3700 D1 673 POP HL
3701 CD C3 30 648 CALL DIV
3702 E5 671 PUSH HL
3703 D5 672 PUSH DE
3704 DD E9 671 JP (IX)
3705 672 672 MOD
3706 DD E1 673 POP IX
3707 E1 674 POP DE
3708 D1 675 POP HL
3709 CD C3 30 648 CALL DIV
3710 E5 673 PUSH HL
3711 D5 674 PUSH DE
3712 DD E9 673 JP (IX)
3713 674 674 MOD
3714 DD E1 675 POP IX
3715 E1 676 POP DE
3716 D1 677 POP HL
3717 CD C3 30 648 CALL DIV
3718 E5 675 PUSH HL
3719 D5 676 PUSH DE
3720 DD E9 675 JP (IX)
3721 676 676 MOD
3722 DD E1 677 POP IX
3723 E1 678 POP DE
3724 D1 679 POP HL
3725 CD C3 30 648 CALL DIV
3726 E5 677 PUSH HL
3727 D5 678 PUSH DE
3728 DD E9 677 JP (IX)
3729 678 678 MOD
3730 DD E1 679 POP IX
3731 E1 680 POP DE
3732 D1 681 POP HL
3733 CD C3 30 648 CALL DIV
3734 E5 679 PUSH HL
3735 D5 680 PUSH DE
3736 DD E9 679 JP (IX)
3737 680 680 MOD
3738 DD E1 681 POP IX
3739 E1 682 POP DE
3740 D1 683 POP HL
3741 CD C3 30 648 CALL DIV
3742 E5 681 PUSH HL
3743 D5 682 PUSH DE
3744 DD E9 681 JP (IX)
3745 682 682 MOD
3746 DD E1 683 POP IX
3747 E1 684 POP DE
3748 D1 685 POP HL
3749 CD C3 30 648 CALL DIV
3750 E5 683 PUSH HL
3751 D5 684 PUSH DE
3752 DD E9 683 JP (IX)
3753 684 684 MOD
3754 DD E1 685 POP IX
3755 E1 686 POP DE
3756 D1 687 POP HL
3757 CD C3 30 648 CALL DIV
3758 E5 685 PUSH HL
3759 D5 686 PUSH DE
3760 DD E9 685 JP (IX)
3761 686 686 MOD
3762 DD E1 687 POP IX
3763 E1 688 POP DE
3764 D1 689 POP HL
3765 CD C3 30 648 CALL DIV
3766 E5 687 PUSH HL
3767 D5 688 PUSH DE
3768 DD E9 687 JP (IX)
3769 688 688 MOD
3770 DD E1 689 POP IX
3771 E1 690 POP DE
3772 D1 691 POP HL
3773 CD C3 30 648 CALL DIV
3774 E5 689 PUSH HL
3775 D5 690 PUSH DE
3776 DD E9 689 JP (IX)
3777 690 690 MOD
3778 DD E1 691 POP IX
3779 E1 692 POP DE
3780 D1 693 POP HL
3781 CD C3 30 648 CALL DIV
3782 E5 691 PUSH HL
3783 D5 692 PUSH DE
3784 DD E9 691 JP (IX)
3785 692 692 MOD
3786 DD E1 693 POP IX
3787 E1 694 POP DE
3788 D1 695 POP HL
3789 CD C3 30 648 CALL DIV
3790 E5 693 PUSH HL
3791 D5 694 PUSH DE
3792 DD E9 693 JP (IX)
3793 694 694 MOD
3794 DD E1 695 POP IX
3795 E1 696 POP DE
3796 D1 697 POP HL
3797 CD C3 30 648 CALL DIV
3798 E5 695 PUSH HL
3799 D5 696 PUSH DE
3800 DD E9 695 JP (IX)
3801 696 696 MOD
3802 DD E1 697 POP IX
3803 E1 698 POP DE
3804 D1 699 POP HL
3805 CD C3 30 648 CALL DIV
3806 E5 697 PUSH HL
3807 D5 698 PUSH DE
3808 DD E9 697 JP (IX)
3809 698 698 MOD
3810 DD E1 699 POP IX
3811 E1 700 POP DE
3812 D1 701 POP HL
3813 CD C3 30 648 CALL DIV
3814 E5 699 PUSH HL
3815 D5 700 PUSH DE
3816 DD E9 699 JP (IX)
3817 700 700 MOD
3818 DD E1 701 POP IX
3819 E1 702 POP DE
3820 D1 703 POP HL
3821 CD C3 30 648 CALL DIV
3822 E5 701 PUSH HL
3823 D5 702 PUSH DE
3824 DD E9 701 JP (IX)
3825 702 702 MOD
3826 DD E1 703 POP IX
3827 E1 704 POP DE
3828 D1 705 POP HL
3829 CD C3 30 648 CALL DIV
3830 E5 703 PUSH HL
3831 D5 704 PUSH DE
3832 DD E9 703 JP (IX)
3833 704 704 MOD
3834 DD E1 705 POP IX
3835 E1 706 POP DE
3836 D1 707 POP HL
3837 CD C3 30 648 CALL DIV
3838 E5 705 PUSH HL
3839 D5 706 PUSH DE
3840 DD E9 705 JP (IX)
3841 706 706 MOD
3842 DD E1 707 POP IX
3843 E1 708 POP DE
3844 D1 709 POP HL
3845 CD C3 30 648 CALL DIV
3846 E5 707 PUSH HL
3847 D5 708 PUSH DE
3848 DD E9 707 JP (IX)
3849 708 708 MOD
3850 DD E1 709 POP IX
3851 E1 710 POP DE
3852 D1 711 POP HL
3853 CD C3 30 648 CALL DIV
3854 E5 709 PUSH HL
3855 D5 710 PUSH DE
3856 DD E9 709 JP (IX)
3857 710 710 MOD
3858 DD E1 711 POP IX
3859 E1 712 POP DE
3860 D1 713 POP HL
3861 CD C3 30 648 CALL DIV
3862 E5 711 PUSH HL
3863 D5 712 PUSH DE
3864 DD E9 711 JP (IX)
3865 712 712 MOD
3866 DD E1 713 POP IX
3867 E1 714 POP DE
3868 D1 715 POP HL
3869 CD C3 30 648 CALL DIV
3870 E5 713 PUSH HL
3871 D5 714 PUSH DE
3872 DD E9 713 JP (IX)
3873 714 714 MOD
3874 DD E1 715 POP IX
3875 E1 716 POP DE
3876 D1 717 POP HL
3877 CD C3 30 648 CALL DIV
3878 E5 715 PUSH HL
3879 D5 716 PUSH DE
3880 DD E9 715 JP (IX)
3881 716 716 MOD
3882 DD E1 717 POP IX
3883 E1 718 POP DE
3884 D1 719 POP HL
3885 CD C3 30 648 CALL DIV
3886 E5 717 PUSH HL
3887 D5 718 PUSH DE
3888 DD E9 717 JP (IX)
3889 718 718 MOD
3890 DD E1 719 POP IX
3891 E1 720 POP DE
3892 D1 721 POP HL
3893 CD C3 30 648 CALL DIV
3894 E5 719 PUSH HL
3895 D5 720 PUSH DE
3896 DD E9 719 JP (IX)
389
```


3283 DD E1	561	POP	IX	3381 6F	727	LD	L,A	3483	895	INSTR2	
3285 3A 08 30	562	LD	A,(*A)	3382 7C	728	LD	A,H	3483 11 00 00	896	LD	DE,0
3288 6F	563	LD	L,A	3383 2F	729	CPL		3486	897	INSTR3	
3289 26 00	564	LD	H,0	3384 67	730	LD	H,A	3486 D5	898	PUSH	DE
328B E5	565	PUSH	HL	3385 01 01 00	731	LD	RC,1	3487 DD E9	899	JP	(IX)
328C DD E9	566	POP	IX	3388 EB	732	EX	DE,HL	3489	900	*STRCMP	
328E	567	ADD	#PUDT	3389 09	733	EX	DE,HL	3489 DD E1	901	POP	IX
328E DD E1	568	POP	IX	338A EB	734	EX	DE,HL	348A E1	902	POP	HL
3290 D1	569	POP	DE	338B 08	735	DEC	BC	348C D1	903	POP	DE
3291 ED 53 09	570	LD	(#DE),DE	338C ED 4A	736	ADC	HL,BC	348D	904	STRCMP1	
3294 30	571	JP	(IX)	338E C9	737	RET		348D 1A	905	LD	A,(DE)
3295 DD E9	572	JP	(IX)	338F	738	STRING ?		348E FE 0D	906	CP	#0D
3297	573	*WORTH		338F	739	POP	IX	348E 2A 0C	907	JP	Z,STRCMP2
3297 DD E1	573	POP	IX	338F	740	POP	HL	348F FE 22	908	CP	
3299 ED 5B 09	574	LD	DE,(#DE)	338F	741	*STRCPY		348A 28 08	909	JP	Z,STRCMP2
329C 30	575	PUSH	DE	338F DD E1	742	POP	IX	348E 46	910	LD	B,(HL)
329D 05	576	JP	(IX)	33C1 E1	743	POP	HL	348F 98	911	SUB	B
32FE DD E9	577	JP	(IX)	33C2 D1	744	POP	DE	348A 28 13	912	JP	NZ,STRCMP3
3300 DD E1	578	POP	IX	33C3	745	STRCPV1		348A 23	913	INC	HL
3302 E1	579	POP	HL	33C3 1A	746	LD	A,(DE)	348B 13	914	INC	DE
3303 22 08 30	580	LD	(#HL),HL	33C4 77	747	LD	(HL),A	348C 18 EF	915	JP	STRCMP1
3306 DD E9	581	JP	(IX)	33C5 FE 0D	748	CP	#0D	348E	916	STRCMP2	
3308	582	*WORTH		33C7 28 0A	749	JP	Z,STRCPY3	348E 01 00 00	917	LD	BC,0
3308 DD E1	583	POP	IX	33C9 FE 22	750	CP	" "	34A1 7E	918	LD	A,(HL)
330A 2A 08 30	584	LD	HL,(#HL)	33C9 28 04	751	JP	Z,STRCPY2	34A2 FE 0D	919	CP	#0D
330D E5	585	PUSH	HL	33C9 28 04	752	INC	HL	34A4 28 07	920	JP	Z,STRCMP4
330E DD E9	586	JP	(IX)	33C9 13	753	INC	DE	34A6 FE 22	921	CP	" "
3310	587			33C9 18 F2	754	JP	STRCPY1	34A8 28 0B	922	JP	Z,STRCMP4
3310 DD E1	591	POP	IX	33D1	755	STRCPY2		34AA 0B	923	DEC	BC
3312 E1	592	POP	HL	33D1 36 0D	756	LD	(HL),#0D	34AB 18 08	924	JP	STRCMP4
3313 E5	593	POP	DE	33D3 DD E9	758	LD	(HL),#0D	34AD 01 01 00	925	LD	BC,1
3314 16 00	594	LD	D,0	33D5	759	*LEFT8		34AD 30 03	927	LD	NC,STRCMP4
3316 05	595	PUSH	DE	33D5 DD E1	760	POP	IX	34B2 01 FF FF	928	LD	BC,-1
3317 DD E9	596	JP	(IX)	33D5 D1	761	POP	HL	34B5	929	PUSH	BC
3319	597	*PERKW		33D9 D1	763	POP	DE	34B5 C5	930	JP	(IX)
3319 DD E1	598	POP	IX	33DA	764	LD	A,(DE)	34B8	931	32 Bit 1-7	
331B E1	599	POP	HL	33DA 1A	765	LD	(HL),A	34B8	932	32 Bit 1-7	
331C 5E	600	LD	E,(HL)	33DB 77	766	CP	#0D	34B8	933	*TASU	
331D 23	601	INC	HL	33DC FE 0D	767	CP	#0D	34B8 DD E1	936	POP	IX
331E 56	602	LD	(HL),D	33DE 28 0D	768	JP	Z,LEFT3	34BA C1	937	POP	BC
331F D5	603	PUSH	DE	33DE 28 0D	769	CP	" "	34BB D1	938	POP	HL
3320 DD E9	604	JP	(IX)	33E4 23	771	INC	HL	34BC E1	939	POP	HL
3322	605	*PORKB		33E5 13	772	INC	DE	34BD 09	940	ADC	HL,BC
3322 DD E1	606	POP	IX	33E6 05	773	DEC	BC	34BD ED	941	EX	HL,C
3324 E1	607	POP	HL	33E8 B1	774	LD	A,B	34BF C1	942	POP	BC
3325 D1	608	POP	DE	33E8 B1	775	OR	C	34C0 ED 4A	943	ADC	HL,BC
3326 F3	609	LD	(HL),E	33E9 28 EF	776	JP	NZ,LEFT1	34C2 E5	944	PUSH	HL
3327 DD E9	610	JP	(IX)	33E9 28 EF	777	LEFT2		34C3 D5	945	PUSH	DE
3329	611	*PORKB		33EB 36 0D	778	LD	(HL),#0D	34CD DD E0	946	JP	(IX)
3329 DD E1	612	POP	IX	33ED	779	LEFT3		34CE	947	*LHUKU	
332B E1	613	POP	HL	33ED DD E9	780	JP	(IX)	34CE DD E1	948	POP	IX
332C D1	614	POP	DE	33EF	781	*RIGHT8		34CE D1	949	POP	DE
332D F3	615	LD	(HL),E	33EF DD E1	782	POP	BC	34C9 C1	950	POP	BC
332E 23	616	INC	HL	33F1 C1	783	POP	BC	34CA E1	951	POP	HL
332F 72	617	LD	(HL),D	33F2 FD E1	784	POP	IX	34CB B7	952	OR	A
3330 DD E9	618	JP	(IX)	33F4 E1	785	POP	HL	34CC ED 52	953	SUB	HL,DE
3332	619	1/0 777		33F5 D5	786	LD	E,L	34CE E1	954	EX	DE,HL
3332	620	1/0 777		33F6 54	787	LD	D,H	34CF E1	955	POP	HL
3332	621			33F7	788	RIGHT1		34D0 ED 42	956	SUB	HL,BC
3332 DD E1	623	POP	IX	33F7 7E	789	LD	A,(HL)	34D2 E5	957	PUSH	HL
3334 C1	624	POP	BC	33F8 FE 22	790	CP	" "	34D3 D5	958	PUSH	DE
3335 DD E5	625	IN	E,(C)	33FA 28 07	791	JP	Z,RIGHT2	34D4 DD E9	959	JP	(IX)
3337 16 00	626	LD	D,0	33FC FE 0D	792	CP	#0D	34DE	960	*LMLT	
3339 D5	627	PUSH	DE	33FE 28 03	793	JP	Z,RIGHT2	34DE DD E1	961	POP	IX
333A DD E9	628	JP	(IX)	3400 23	794	INC	HL	34DE D1	962	POP	HL
333C	629	*WOUT		3401 18 F4	795	JP	RIGHT1	34D9 E1	963	POP	HL
333C DD E1	630	POP	IX	3403	796	RIGHT2		34DA D9	964	EXX	HL
333E E1	631	POP	HL	3403 2B	797	DEC	HL	34DB FD 21 00	965	LD	IV,0
333F C1	632	POP	BC	3404 E5	798	PUSH	HL	34DE 21 00 00	966	LD	HL,0
3340 DD E9	633	OUT	(C),L	3405 B7	799	OR	A	34E2 D1	967	POP	DE
3342 DD E9	634	JP	(IX)	3406 ED 52	800	SBC	HL,DE	34E2 D1	968	POP	BC
3344	635			3408 E1	801	POP	HL	34E4 D9	969	EXX	BC
3344	636			3408 08	803	DEC	BC	34E5 06 20	970	LD	B,32
3344	637	*HIGH		340C 7A	804	LD	A,B	34E7	971	LMLT1	
3344 DD E1	638	POP	IX	340D B1	805	OR	C	34E7 CB 3C	972	SERL	H
3346 E1	639	POP	HL	340E 28 F3	806	JP	NZ,RIGHT2	34E9 CB 1D	973	RR	L
3347 8C	640	LD	L,H	3410 FD E5	808	RIGHT3		34EB CB 1A	974	RR	E
3348 26 00	641	LD	H,0	3412 D1	809	POP	DE	34ED CB 1B	975	RR	E
334A E5	642	PUSH	HL	3413 3B	810	POP	HL	34EF D9	976	EXX	
334B DD E9	643	JP	(IX)	3414 18 AD	811	JP	STRCPV1	34F0 38 04	977	JP	NC,LMLT2
334D	644	*LOW		3416	812	*MID5		34F2 FD 19	978	ADC	HL,BC
334D DD E1	645	POP	IX	3416 DD E1	813	POP	IX	34F4 ED 4A	979	ADC	HL,BC
334F E1	646	POP	HL	3418 D9	814	EXX		34F6	980	LMLT2	
3350 26 00	647	LD	H,0	3419 C1	815	POP	BC	34F6 CB 23	981	SLA	E
3352 E5	648	PUSH	HL	341A D9	816	EXX		34F8 CB 12	982	RL	D
3353 DD E9	649	JP	(IX)	341B C1	817	POP	BC	34FA CB 10	983	RL	C
3355	650	*EX		341C E1	818	POP	HL	34FC CB 10	984	RL	B
3355 DD E1	651	POP	IX	341D D1	819	POP	DE	34FE D9	985	EXX	
3357 E1	652	POP	HL	341D D1	819	POP	DE	34FF 19 E6	986	DJNZ	LMLT1
3358 70	653	LD	A,L	341E	820	MID1		3501 D9	987	EXX	
3359 6C	654	LD	L,H	341E 08	821	DEC	BC	3502 E5	988	PUSH	HL
335A 67	655	LD	R,A	341F 79	822	LD	A,B	3503 FD E5	989	PUSH	IX
335B E5	656	PUSH	HL	3420 B1	823	OR	C	3505 DD E9	990	JP	(IX)
335C DD E9	657	JP	(IX)	3421 28 08	824	JP	Z,MID2	3507	991	*LDIV	
335E	658	*WNOT		3423 16	825	LD	A,(DE)	3507 DD E1	992	POP	IX
335E DD E1	659	POP	IX	3424 FE 22	826	CP	" "	3509 CD 2C 35	993	CALL	LDIV
3360 E1	660	POP	HL	3428 FE 0D	827	CP	#0D	350C D9	994	EXX	
3361 7D	661	LD	A,L	3428 13	828	JP	MID1	350D C5	995	PUSH	BC
3362 2F	662	CPL		3429 18 F3	829	JP	MID1	350E D9	996	PUSH	BC
3363 6F	663	LD	L,A	342B	830	MID2		350F C5	997	JP	(IX)
3364 7C	664	LD	A,H	342B D9	831	EXX		3510 DD E9	998	EXX	
3365 2F	665	CPL		342C C5	832	PUSH	BC	3512	999	*LMOD	
3366 67	666	LD	H,A	342D 09	833	EXX		3512 DD E1	1000	POP	IX
3367 BC	667	PUSH	HL	342D C1	834	POP	BC	3514 CD 2C 35	1001	CALL	LDIV
3368 DD E9	668	JP	(IX)	342E 09	835	MID3		3517 D9	1002	EXX	
336A	669	*ROR		342F 1A	836	LD	A,(DE)	3518 E5	1003	PUSH	HL
336A DD E1	670	POP	IX	3430 77	837	LD	(HL),A	3519 D9	1004	EXX	
336C C1	671	POP	BC	3431 FE 22	838	CP	" "	351A E5	1005	PUSH	HL
336D E1	672	POP	HL	3433 28 08	839	JP	Z,MID4	351B DD E9	1006	JP	(IX)
336E 41	673	LD	B,C	3435 FE 0D	840	CP	#0D	351D DD E1	1007	POP	IX
336F	674	*RORI		3437 2F 27	841	JP	Z,MID4	351F CD 2C 35	1008	CALL	LDIV
336F CB 3C	675	SRL	H	3439 23	842	INC	HL	3520 D9	1009	PUSH	HL
3371 CB 1D	676	RR	L	343A 13	843	INC	DE	3523 E5	1011	PUSH	HL
3373 10 FA	677	C	DJNZ RORI	343B 09	844	DEC	BC	3524 D9	1012	EXX	
3375 E5	678	LD	H,0	343C 78	845	LD	A,B	3525 E5	1013	PUSH	HL
3376 DD E9	679	JP	(IX)	343D B1	846	OR	C	3526 D9	1014	EXX	
3378	680	*RORI		343E 28 EF	847	JP	NZ,MID3	3527 C5	1015	PUSH	BC
3378 DD E1	681	POP	IX	3440	848	MID4		3528 D9	1016	EXX	
337A C1	682	POP	BC	3440 36 0D	849	LD	(HL),#0D	3529 C5	1017	PUSH	BC
337B E1	683	POP	HL	3442 DD E9	850	JP	(IX)	352A DD E9	1018	JP	(IX)
337C 41	684	LD	B,C	3444	851	*STRCAT		352C	1019		
337D	685	*ROL1		3444 DD E1	852	POP	IX	352C	1020	BC=BC/BC/DE/DE	
337D 29											

3560 1063 LDIV3 POP AF
3561 1064 POP AF
3562 20 DA 1066 JR NZ,LDIV1
3564 FD E9 1067 JP (IX)
3566 1068 LDIV2
3566 19 1069 ADD HL,DE
3567 D9 1070 EXX
3568 ED 5A 1071 ADC HL,DE
356A D9 1072 EXX
356B 18 F3 1073 JR LDIV3
356D 1074
356D DD E1 1075 POP IX
356F C1 1076 POP BC
3570 D1 1077 POP DE
3571 E1 1078 POP HL
3572 CD 7C 35 1079 CALL QUOT
3575 D9 1080 EXX
3576 D9 1081 PUSH DE
3577 D9 1082 EXX
3578 E5 1083 PUSH HL
3579 D5 1084 PUSH DE
357A DD E9 1085 JP (IX)
357C 1086
357C 1087 HLDE=HLDE/BC
357C 1088
357C 1089 DE'=HLDE MOD BC
357C 1090
357C 1091 QUOT
357C F5 1092 PUSH AF
357D C5 1093 PUSH BC
357E D9 1094 EXX
357F C1 1095 POP BC
3580 21 00 00 1096 LD HL,0
3583 11 00 00 1097 LD DE,0
3586 D9 1098 EXX
3587 3E 20 1099 LD A,32
3589 1100 QUOT1
358A 29 1101 EX DE,HL
358B 29 1102 ADD HL,HL
358D EB 1103 EX DE,HL
358E DD 6A 1104 ADC HL,HL
358F EB 1105 EXX
3590 ED 6A 1106 EX DE,HL
3592 EB 1107 ADC HL,HL
3593 D5 1108 PUSH HL
3594 E5 1109 ADC HL,HL
3595 E5 1110 PUSH DE
3596 67 1111 PUSH HL
3597 CD AA 35 1112 CALL QUOT8B
359A E1 1113 POP HL
359B D1 1114 POP DE
359C 38 E3 1115 JR C,QUOT2
359D CD AA 35 1116 CALL QUOT8B
35A1 1117
35A1 D9 1118 EXX
35A2 38 E1 1119 JR C,QUOT3
35A4 1C 1120 INC E
35A5 1121 QUOT3
35A5 3D 1122 DEC A
35A6 20 E1 1123 JR NZ,QUOT1
35A8 F1 1124 POP AF
35A9 C9 1125 RET
35AA 1126 QUOT8B
35AA EB 1127 EX DE,HL
35AB B7 1128 OR A
35AC ED 42 1129 SBC HL,BC
35AD EB 1130 EX DE,HL
35AF D6 1131 RET NC
35B0 67 1132 LD H,A
35B1 7D 1133 LD A,L
35B2 D6 E1 1134 SUB 1
35B4 67 1135 LD L,A
35B5 7C 1136 LD H,A
35B6 3E 00 1137 LD A,0
35B8 C9 1138 RET
35B9 1139 DMULT
35B9 DD E1 1140 POP IX
35B9 D1 1141 POP BC
35B9 D1 1142 POP DE
35B9 D1 1143 LD HL,0
35C0 D9 1144 EXX
35C1 11 00 00 1145 LD DE,0
35C4 21 00 00 1146 LD HL,0
35C7 D9 1147 EXX
35C8 3E 10 1148 LD A,16
35CA 3C 38 1149 SBC HL,BC
35CB C9 19 1150 RR C
35CC 3E 85 1151 JR NC,DMULT2
35CE 19 1152 ADD HL,DE
35D1 D9 1153 EXX
35D2 ED 5A 1154 ADC HL,DE
35D4 D9 1155 EXX
35D5 1156 DMULT2
35D5 CD 23 1157 SBC HL,DE
35D7 CD 12 1158 EXX
35D9 D9 1159 EXX
35DA CD 13 1160 RL E
35DC CD 12 1161 RL D
35DE D9 1162 EXX
35DF 30 1163 EXX
35E0 D9 1164 DEC A
35E2 20 E8 1165 JR NZ,DMULT1
35E2 D9 1166 EXX
35E3 E5 1167 PUSH HL
35E4 D9 1168 EXX
35E5 E5 1169 PUSH HL
35E6 DD E9 1170 JP (IX)
35E8 1171 DMULT1
35E8 DD E1 1172 POP IX
35EA C1 1173 POP BC
35EB D1 1174 POP DE
35EC 21 00 00 1175 LD HL,A
35EF 79 1176 LD A,C
35F0 1177 HLIT1
35F0 B7 1178 OR A
35F1 20 0C 1179 JR 2,HLIT3
35F3 CD 3F 1180 SBC HL,DE
35F5 30 01 1181 JR NC,HLIT2
35F7 19 1182 ADD HL,DE
35F8 1183 HLIT2
35F8 CD 23 1184 SBC HL,DE
35FA CD 12 1185 RL D
35FC CD 35 1186 JP HLIT1
35FF 1187 HLIT3
35FF E5 1188 PUSH HL
3580 DD E9 1189 JP (IX)
3582 1190 CMPT2
3582 DD E1 1191 POP IX
3584 C1 1192 POP BC
3585 D9 1193 POP DE
3586 E1 1194 POP HL
3587 B7 1195 OR A
3588 ED 42 1196 SBC HL,BC
358A 4D 1197 LD C,L
358B 44 1198 LD B,H
358C E1 1199 POP HL
358D ED 52 1200 SBC HL,DE
358F 38 09 1201 JR C,CMPT2_1
3591 7C 1202 LD A,H
3592 85 1203 OR L
3593 D6 1204 OR D
3594 B1 1205 OR C
3595 28 08 1206 JR NZ,CMPT2_2
3597 1207 CMPT2_3
3597 E5 1208 PUSH HL
3598 DD E9 1209 JP (IX)
359A 1210 CMPT2_1
359A 1D FF FF 1211 LD HL,-1
359D 18 F8 1212 JR CMPT2_3
359F 21 01 00 1213 LD HL,1
35A2 18 F3 1214 JR CMPT2_3
35A4 1215 CMPT2_1
35A4 DD E1 1216 POP IX
35A6 E1 1217 POP HL
35A7 11 00 00 1218 LD DE,0
35A8 CD 7C 1219 DJT 7,H
35A9 28 01 1220 JR 2,CTL1
35AB 28 01 1221 DEC DE
35AC 1222 CTL1
35AD D5 1223 PUSH DE
35AE E5 1224 PUSH DE
35AF DD E9 1225 JP (IX)
35B0 1226 JASCII
35B3 DD E1 1227 POP IX
35B5 E1 1228 POP HL
35B6 11 00 00 1229 LD DE,0
35B9 1230 ASCI11

3539 7E 1232 LD A,(HL)
353A 23 1233 INC HL
353B 0E 0D 1234 CP #00
353C 28 08 1235 JR Z,ASCI12
353F FE 22 1236 CP #1
3541 28 04 1237 JR Z,ASCI12
3543 53 1238 LD D,E
3544 5F 1239 LD E,A
3545 18 F2 1240 JR ASCI11
3547 1241 ASCI12
3547 D5 1242 PUSH DE
3548 DD E9 1243 JP (IX)
354A 1244 POP C
354A DD E1 1245 POP IX
354C 1246 POP DE
354D E1 1247 POP HL
354E 01 01 00 1248 LD BC,1
3551 CB 7A 1249 BIT 7,D
3553 20 0D 1250 JR NZ,F4
3555 CB 7C 1251 BIT 7,H
3557 20 06 1252 JR NZ,F4END
3559 1253 F4
3559 B7 1254 OR A
355A DD 52 1255 SBC HL,DE
355C 38 01 1256 JR C,F4END
355E 1257 F43
355E 06 1258 DEC BC
355F 1259 F4END
355F C5 1260 PUSH BC
356A DD E9 1261 JP (IX)
356A 1262 F41
356C CB 7C 1263 BIT 7,H
356A 28 F8 1264 JR Z,F43
356E 18 F1 1265 JR F4
3568 1266
3568 DD E1 1267 POP IX
356A E1 1268 POP HL
356A E1 1269 POP HL
356B 11 01 00 1270 LD DE,1
356B 7C 1271 LD A,A
356F B5 1272 OR L
3570 28 E1 1273 JR 2,001
3572 1D 1274 DEC D
3573 D5 1275 001
3573 D5 1276 PUSH DE
3574 DD E9 1277 JP (IX)
3576 1278 01C#
3576 DD E1 1279 POP IX
357A E1 1280 POP HL
3579 23 1281 INC HL
357A E5 1282 PUSH HL
3578 DD E9 1283 JP (IX)
357D 1284 0DEC#
357D DD E1 1285 POP IX
357E E1 1286 POP HL
3580 2B 1287 DEC HL
3581 E5 1288 PUSH HL
3582 DD E9 1289 JP (IX)
3584 1290 0PRINT1
3584 DD E1 1291 POP IX
3586 D1 1292 POP DE
3587 20 00 37 1293 LD HL,0
358A CD A0 37 1294 CALL CVHLDE
358D 11 C4 37 1296 LD DE,0CVBUFB
358E DD E9 1297 CALL #MSG
358F E5 1298 JP (IX)
3595 1299 0PRINT2
3595 DD E1 1300 POP IX
3597 1301 POP DE
3598 E1 1302 POP HL
3599 18 EF 1303 JR 0PRINT1
359A DD E1 1304 0PRV
359D DD E1 1305 POP IX
359D E1 1306 POP DE
359E 00 00 1307 LD DE,0
35A1 CD 7C 1308 BIT 7,H
35A3 28 08 1309 JR 2,PRINTF1
35A5 3E 2D 1310 LD A,-1
35A7 CD F4 1F 1311 CALL #PRINT
35AA CD A1 33 1312 CALL NEGATE
35AD 1313 PRINTF1
35AD EB 1314 EX DE,HL
35AE 18 DA 1315 JR 0PRINT1
35B0 1316 0PWF2
35B0 DD E1 1317 POP IX
35B2 D1 1318 POP DE
35B3 E1 1319 POP HL
35B4 CD 7C 1320 BIT 7,H
35B6 28 D2 1321 JR Z,0PRINT1
35B8 2D 1322 LD A,-1
35BA CD F4 1F 1323 CALL #PRINT
35BD CD A9 33 1324 CALL NEGATE2
35C0 CD 8A 36 1325 JP 0PRINT1
35C3 1326 0STRV
35C3 DD E1 1327 POP IX
35C5 E1 1328 POP HL
35C6 D1 1329 POP DE
35C7 E5 1330 PUSH HL
35C8 21 00 00 1331 LD HL,0
35C8 1332 0STRW1
35C8 CD A0 37 1333 CALL CVHLDE
35CE 11 C4 37 1334 LD DE,0CVBUFB
35D2 C3 33 1336 JP STRCPY1
35D5 1337 0STRL
35D5 DD E1 1338 POP IX
35D7 C1 1339 POP BC
35D8 D1 1340 POP DE
35D9 E1 1341 POP HL
35DA C5 1342 PUSH BC
35DB 18 EE 1343 JR STRW1
35DD DD E1 1344 0HEXL
35DE DD E1 1345 POP IX
35DF D1 1346 POP DE
35E0 E1 1347 POP HL
35E1 CD BE 1F 1348 CALL #PRTHL
35E4 E8 1349 EX DE,HL
35E5 CD BE 1F 1350 CALL #PRTHL
35E8 DD E9 1351 JP (IX)
35EA 1352 0VAL1
35EA DD E1 1353 POP IX
35EC FD E1 1354 POP DE
35EE FD 7E 00 1355 LD A,(IV)
35F1 FE 2D 1356 CP -
35F3 28 06 1357 JR 2,0VAL1_1
35F5 CD D4 37 1358 CALL DECI
35F6 1359 0VAL1_2
35F8 E5 1360 PUSH HL
35F9 DD E9 1361 JP (IX)
35FA 1362 0VAL1_1
35FB FD 23 1363 INC Y
35FC DD 03 1364 CALL DECI
35FD CD A1 33 1365 CALL NEGATE
35FE 18 F3 1366 JR 0VAL1_2
35FE 1367 0VAL2
35FE DD E1 1368 POP IX
35FF FD E1 1369 POP DE
35FF CD EE 37 1370 CALL HLDEDECI
35FF E5 1371 PUSH HL
35FF D5 1372 PUSH DE
35FF DD E9 1373 JP (IX)
35FF 1374 0VAL3
35FF DD E1 1375 POP IX
35FF FD E1 1376 POP DE
35FF CD 07 38 1377 CALL #HEX
35FF E5 1378 PUSH HL
35FF DD E9 1379 JP (IX)
35FF 1380 0INP2
35FF DD E1 1381 POP IX
35FF CD #CSR 1382 LD DE,0
35FF 2E 00 1383 LD E,(#KRFAD)
35FF 2D 5B 76 1384 LD DE,0
35FF 1385 CALL #GRTL
35FF CD D3 1F 1386 LD A,(DE)
35FF 28 18 1387 CP B1
35FF 28 0C 1388 JR Z,INP3
35FF 2D 19 1389 ADD HL,DE
35FF 28 08 1390 EX DE,HL
35FF E1 1391 POP IX
35FF 1392 INP2
35FF 1393 LD A,(DE)
35FF B7 1394 OR A
35FF 28 06 1395 JR Z,INP3
35FF 1396 LD (HL),A
35FF 13 1397 INC DE
35FF 23 1398 INC HL
35FF 18 F7 1399 JR INP2

3739 1400 INP1 POP HL
3739 E1 1401 POP HL
373A 1402 INP3 LD (HL),00D
373A 36 0D 1403 JR Z,ASCI12
373C DD E9 1404 JP (IX)
373E 1405 0TRANS1
373E DD E1 1406 POP IX
373E DD E1 1407 POP BC
3741 D1 1408 POP DE
3742 E1 1409 POP HL
3743 ED B0 1410 LDIR
3743 DD E9 1411 JP (IX)
3747 1412 0TRANS2
3747 DD E1 1413 POP IX
3749 C1 1414 POP BC
374A D1 1415 POP DE
374B E1 1416 POP HL
374C DD B8 1417 LDOR
374E DD E9 1418 JP (IX)
3750 1419 0FTLL
3750 DD E1 1420 POP IX
3752 D1 1421 POP DE
3753 C1 1422 POP BC
3754 E1 1423 POP HL
3755 08 1424 DEC BC
3756 73 1425 LD (HL),E
3757 54 1426 LD D,H
3758 DD 1427 LD E,L
3759 13 1428 INC DE
375A ED B0 1429 LDOR
375C DD E9 1430 LDIR
375E 1431 0COPYL
375E DD E1 1432 POP IX
3760 D1 1433 POP DE
3761 E1 1434 POP HL
3762 E5 1435 PUSH HL
3763 D5 1436 PUSH DE
3764 E5 1437 PUSH HL
3765 D5 1438 PUSH DE
3766 DD E9 1439 JP (IX)
3768 1440 0DOPL
3768 DD E1 1441 POP IX
3768 E1 1442 POP HL
376B E1 1443 POP DE
376C DD E9 1444 JP (IX)
376E 1445 0SWAPD
376E DD E1 1446 POP IX
3770 D1 1447 POP DE
3771 E1 1448 POP HL
3772 DD E9 1449 JP (IX)
3773 D1 1450 POP DE
3774 E1 1451 POP HL
3775 DD E1 1452 PUSH HL
3776 E5 1453 PUSH DE
3777 D5 1454 PUSH HL
3778 D5 1455 PUSH DE
3779 E5 1456 PUSH HL
377A D5 1457 PUSH DE
377B DD E9 1458 JP (IX)
377D 1459 0BREA
377D CD 1F 1460 CALL BREA
3780 C1 1461 RET
3781 C3 52 32 1462 EQU #1F09
3784 1463 EQU #1F0E
3784 1464 EQU #1F0F
3784 1465 EQU #1F0E
3784 1466 EQU #1F0E
3784 1467 EQU #1F0E
3784 1468 EQU #1F0E
3784 C3 D9 1F 1469 JP #LPTON
3787 1470 0PROFF
3787 C3 D6 1F 1471 JP #LPTOP
378A 1472 0PEEK
378A DD E1 1473 POP IX
378C E1 1474 POP HL
378D CD 94 1F 1475 CALL #PEEK
378F 6F 1476 LD A,C
3791 26 08 1477 LD H,0
3793 E5 1478 PUSH HL
3794 DD E9 1479 JP (IX)
3796 1480 0PORA
3796 DD E1 1481 POP IX
3798 C1 1482 POP DE
3799 E1 1483 POP HL
379A 79 1484 LD A,C
379B CD 0A 1F 1485 CALL #PORA
379E DD E9 1486 JP (IX)
37A0 1487
37A0 1488 0VHLDE
37A0 01 00 00 1489 LD BC,0
37A3 1490 0VHLDE1
37A3 C5 1491 PUSH BC
37A4 01 0A 00 1492 LD BC,10
37A7 CD 7C 35 1493 CALL QUOT
37AA DD E9 1494 EXX
37AB 3E 30 1495 LD A,'0'
37AD B3 1496 ADD A,E
37AE DD E9 1497 EXX
37AF C1 1498 POP BC
37B0 03 1499 INC BC
37B1 F5 1500 PUSH AF
37B2 7C 1501 LD H,H
37B3 B5 1502 OR L
37B4 D2 1503 OR D
37B5 B3 1504 OR C
37B6 20 B8 1505 JR NZ,CVHLDE1
37B8 4D 1506 LD B,C
37B9 21 C4 37 1507 LD HL,CVBUFB
37BC 1508 0VHLDE2
37BC F1 1509 POP AF
37BD 77 1510 LD (HL),A
37BE 23 1511 INC HL
37BF 18 FB 1512 DJNZ CVHLDE2
37C1 36 0D 1513 LD (HL),00D
37C3 C9 1514 RET
37C4 1515 0CVBUFB
37C4 00 00 00 1516 DS 12
37D0 1517
37D0 1518 ; CONVERT DECIMAL TO HL
37D0 1519
37D0 1520 DEC1
37D0 21 00 00 1521 LD HL,0
37D3 DD 7E 00 1523 LD A,(IV)
37D6 FE 30 1524 CP '0'
37D8 DD 00 1525 RET C
37D9 FE 3A 1526 CP '9'+1
37DA DD 00 1527 RET NC
37DB DD 23 1528 INC Y
37DE 29 1529 ADD HL,HL
37DF 54 1530 LD D,H
37E0 5D 1531 LD E,L
37E1 29 1532 ADD HL,HL
37E2 29 1533 ADD HL,HL
37E3 19 1534 ADD HL,DE
37E4 D6 30 1535 SUB '0'
37E5 B5 1536 ADD A,L
37E7 30 01 1537 JR NC,DEC12
37E9 24 1538 INC H
37FA 1539 DEC12
37FA 6F 1540 LD L,A
37FB C3 D3 37 1541 JP DEC11
37FE DD 00 1542 0HLDEDEC1
37FE 11 00 00 1543 LD HL,0
37FE 11 00 00 1544 LD DE,0
37FA FD 7E 00 1545 HLDE_1
37FF FE 30 1546 LD A,(IV)
37FF DD 00 1547 CP '0'
37FF DD 3A 1548 RET C
37FF D6 30 1549 CP '9'+1
37FF DD 00 1550 RET NC
37FF DD 23 1551 INC Y
37FF D6 30 1552 SUB '0'
37FF DD 00 1553 EX DE,HL ; HLDE *2
37FF DD 23 1554 ADD HL,HL
37FF DD 00 1555 ADD HL,HL
37FF DD 6A 1556 ADC HL,HL
37FF 1557
37FF 1558 LD B,H
37FF DD 00 1559 LD C,L
37FF DD 00 1560 PUSH BC
37FF DD 00 1561 LD B,D
37FF DD 00 1562 LD C,E
37FF 1563
37FF DD 00 1564 EX DE,HL
37FF DD 00 1565 ADD HL,HL

3958 36 02	1773	LD	(HL),2
3958 06 04	1774	LD	(HL),1
395A 18 D5	1775	JM	WDATA
395C	1776	WSLINE	
395C PD E1	1777	POP	Y
395E 21 F3 39	1778	LD	HL,MAGICBUF
3961 36 01	1779	LD	(HL),2
3964 06 06	1780	LD	B,4
396E 18 CA	1781	JM	WDATA
3967	1782	WBOX	
3967 PD E1	1783	POP	Y
3969 21 F3 39	1784	LD	HL,MAGICBUF
396E 36 02	1785	LD	(HL),2
396E 06 04	1786	LD	B,4
3970 18 BF	1787	JM	WDATA
3972	1788	WFILE	
3972 PD E1	1789	POP	Y
3974 21 85 39	1790	LD	HL,TILEBUF+3
3977 D1	1791	POP	DE
3977 12 02	1792	LD	(HL),D
3979 2B	1793	DEC	HL
397A 73	1794	LD	(HL),E
397A 01	1795	POP	DE
397C 2B	1796	DEC	HL
397D 72	1797	LD	(HL),D
397E 1D 00	1798	DEC	HL
397F 73	1799	LD	(HL),E
3980 PD E9	1800	JP	(IY)
3982	1801	TILEBUF	
3982 FF FF	1802	DW	FFFF
3984 FF FF	1803	DW	FFFF
3986	1804	WBOXFUL	
3986 PD E1	1805	POP	Y
3988 21 F3 39	1806	LD	HL,MAGICBUF
398E 36 04	1807	LD	(HL),4
398D 06 04	1808	LD	B,4
398F	1809	BOXFI	
398F 23	1810	INC	HL
3990 C5	1811	PUSH	BC
3991 11 82 39	1812	LD	DE,TILEBUF
399A 01 04 00	1813	LD	BC,4
3997 EB	1814	EX	DE,HL
3999 DD D0	1815	LDIR	EX
399A EB	1816	EX	DE,HL
399B C1	1817	POP	BC
399C 2B	1818	DEC	HL
399D 16 92	1819	JM	WDATA
399F	1820	WTRIANGLE	
399F PD E1	1821	POP	Y
399A 21 F3 39	1822	LD	HL,MAGICBUF
39A4 36 03	1823	LD	(HL),3
39A6 06 06	1824	LD	B,6
39A8 16 85	1825	JM	BOXFI
39AA	1826	WCIRCLE	
39AA PD E1	1827	POP	Y
39A9 21 F3 39	1828	LD	HL,MAGICBUF
39AB 36 05	1829	LD	(HL),5
39B1 06 03	1830	LD	B,3
39B3 18 DA	1831	JM	BOXFI
39B5	1832	WDOT	
39B5 PD E1	1833	POP	Y
39B7 21 F3 39	1834	LD	HL,MAGICBUF
39BA 36 00	1835	LD	(HL),0
39BC 23	1836	INC	HL
39BD 36 01	1837	LD	(HL),1
39BF 06 02	1838	LD	B,2
39C1 C3 31 39	1839	JM	WDATA
39C4	1840	WPHAGIC	
39C4 PD E1	1841	POP	Y
39C6 DD E1	1842	POP	IX
39C8 C3 84 38	1843	JP	INITI
39CB	1844	WPHPOINT	
39CB PD E1	1845	POP	Y
39CD 21 F3 39	1846	LD	HL,MAGICBUF
39D0 36 08	1847	LD	(HL),8
39D2 C1	1848	POP	BC
39D3 21	1849	POP	DE
39D4 23	1850	INC	HL
39D5 73	1851	LD	(HL),E
39D6 12	1852	INC	HL
39D7 72	1853	LD	(HL),D
39D8 23	1854	INC	HL
39D9 71	1855	LD	(HL),C
39DA 23	1856	INC	HL
39DB 78	1857	LD	(HL),B
39DC 23	1858	INC	HL
39DD 36 0F	1859	LD	(HL),80F
39DE PD E5	1860	PUSH	Y
39E0 21 F3 861	1861	LD	IX,MAGICBUF
39E4 39			
39E5 CD 04 B0	1862	CALL	MAGIC
39E8 PD E1	1863	POP	Y
39FA 3A 02 C2	1864	LD	A,(\$C20)
39ED 6F	1865	LD	L,A
39EE 36 00	1866	POP	DE
39F0 E5	1867	PUSH	HL
39F1 PD E9	1868	JP	(IY)
39F3	1869	MAGICBUF	
39F3 00 00 00	1870	DS	32
39F5 00 00 00			
39F9 00 00 00			
39FC 00 00 00			
39FF 00 00 00			
3A02 00 00 00			
3A05 00 00 00			
3A08 00 00 00			
3A0B 00 00 00	</		

3AF0	CF	68	RET Z				
3AF1	BC	69	LD	BC,HL			
3AF2	BC	70	EX	DE,HL	EX	DE,HL	
3AF3	BC	71	JP	0000			
3AF4		72	MAIN				
3AFA	CD	73	CALL	#ERR1			
3AFD	ED	74	DM	PASS 1"			
3B00	53	20	31				
3B03	00	00	75	DE	\$00,0		
3B05	CD	45	3B	76	CALL	CMAIN	
3B08	CD	42	3F	77	CALL	#ERR1	
3B0A	ED	41	53	78	DM	PASS 2"	
3B0E	53	22	1F	79			
3B11	00	00	70	DB	\$00,0		
3B13	CD	45	3B	80	CALL	CMAIN	
3B16	CD	42	3F	81	CALL	#ERR1	
3B19	4F	42	4A	82	DM	OBJECT END:"	
3B1C	45	43	54				
3B1F	20	45	54				
3B22	44	3A	80				
3B24	00	00	83	DB	\$00,0		
3B27	51	52	32	84	LD	HL,(END)	
3B29	5F	36	00	85	LD	HL,\$03	
3B2C	33						
3B2D	5F	75	01	86	LD	(IY+1),L	
3B30	5F	74	02	87	LD	(IY+2),H	
3B33	5F	73	88		INC	YI	
3B35	5F	73	89		INC	YI	
3B37	5F	73	89		INC	YI	
3B39	5F	E5	91		PUSH	YI	
3B3B	E1		92		POP	HL	
3B3C	CD	BE	1F	93	CALL	#ERR1	
3B3F	CD	EB	1F	94	CALL	NL	
3B42	53	FA	1F	95	JP	FFFA	
3B45			96	CCMAIN			
3B4F	AF				XOR	A	
3B50	52	90	3E	98	LD	(IFCUNT),A	
3B53	0D	2A	91	99	LD	IX,(TEXT)	
3B5C							
3B6D	5F	2A	93	100	LD	YI,(OBJECT)	
3B69	5E						
3B71	ED	4B	9B	101	LD	BC,(OFFSET)	
3B74	5E						
3B75	5F	00	99	102	LD	YI,BC	
3B77	21	0D	30	103	LD	HL,\$START	
3B7A	5F	36	00	104	LD	(IY),BC	
3B7D	CD						
3B7E	5F	75	01	105	LD	(IY+1),L	
3B80	5F	74	02	106	LD	(IY+2),H	
3B84	5F	73	88		INC	YI	CALL \$START
3B86	5F	73	89		INC	YI	
3B88	5F	73	89		INC	YI	
3B8A	5F	73	89		INC	YI	
3B8D	5F	36	00	110	LD	(YI),#ED	LD (MAIN_SP),SP


```

3B6V FD 36 01 111 LD (IY+1),#73
3B71 73 LD HL,MAIN_SP
3B72 21 00 30 112 LD (IY+2),L
3B73 75 02 113 LD (IY+3),H
3B74 74 03 114 LD (IY+4),H
3B75 76 04 115 LD (IY+5),H
3B76 31 LD HL,(INT_TOP)
3B77 2A 97 116 LD HL,(RET_TOP)
3B78 75 05 117 LD (IY+6),H
3B79 74 06 118 LD (IY+7),H
3B80 76 07 119 LD HL,(RET_TOP)
3B81 21 LD HL,(RET_TOP)
3B82 2A 99 3E 120 LD (IY+8),L
3B83 75 08 121 LD (IY+9),H
3B84 74 09 122 LD (IY+10),H
3B85 76 0A 123 LD (IY+11),H
3B86 22 LD HL,(RET_SF),HL
3B87 21 02 30 124 LD HL,RET_SF
3B88 75 0B 125 LD (IY+12),L
3B89 74 0C 126 LD (IY+13),H
3B90 76 0D 127 LD HL,RET_SF
3B91 21 02 30 128 LD HL,RET_SF
3B92 75 0E 129 LD (IY+14),L
3B93 74 0F 130 LD (IY+15),H
3B94 76 10 131 LD HL,RET_SF
3B95 21 02 30 132 LD HL,RET_SF
3B96 75 11 133 LD (IY+16),L
3B97 74 12 134 LD (IY+17),H
3B98 76 13 135 LD HL,RET_SF
3B99 21 02 30 136 LD HL,RET_SF
3B9A 75 14 137 LD (IY+18),L
3B9B 74 15 138 LD (IY+19),H
3B9C 76 16 139 LD HL,RET_SF
3B9D 21 02 30 140 LD HL,RET_SF
3B9E 75 17 141 LD (IY+20),L
3B9F 74 18 142 LD (IY+21),H
3BA0 76 19 143 LD HL,RET_SF
3BA1 21 02 30 144 LD HL,RET_SF
3BA2 75 1A 145 LD (IY+22),L
3BA3 74 1B 146 LD (IY+23),H
3BA4 76 1C 147 LD HL,RET_SF
3BA5 21 02 30 148 LD HL,RET_SF
3BA6 75 1D 149 LD (IY+24),L
3BA7 74 1E 150 LD (IY+25),H
3BA8 76 1F 151 LD HL,RET_SF
3BA9 21 02 30 152 LD HL,RET_SF
3BAB 75 20 153 LD (IY+26),L
3BAC 74 21 154 LD (IY+27),H
3BAD 76 22 155 LD HL,RET_SF
3BAE 21 02 30 156 LD HL,RET_SF
3BAF 75 23 157 LD (IY+28),L
3BB0 74 24 158 LD (IY+29),H
3BB1 76 25 159 LD HL,RET_SF
3BB2 21 02 30 160 LD HL,RET_SF
3BB3 75 26 161 LD (IY+30),L
3BB4 74 27 162 LD (IY+31),H
3BB5 76 28 163 LD HL,RET_SF
3BB6 21 02 30 164 LD HL,RET_SF
3BB7 75 29 165 LD (IY+32),L
3BB8 74 2A 166 LD (IY+33),H
3BB9 76 2B 167 LD HL,RET_SF
3BBA 21 02 30 168 LD HL,RET_SF
3BBB 75 2C 169 LD (IY+34),L
3BBC 74 2D 170 LD (IY+35),H
3BBD 76 2E 171 LD HL,RET_SF
3BBE 21 02 30 172 LD HL,RET_SF
3BBF 75 2F 173 LD (IY+36),L
3BC0 74 30 174 LD (IY+37),H
3BC1 76 31 175 LD HL,RET_SF
3BC2 21 02 30 176 LD HL,RET_SF
3BC3 75 32 177 LD (IY+38),L
3BC4 74 33 178 LD (IY+39),H
3BC5 76 34 179 LD HL,RET_SF
3BC6 21 02 30 180 LD HL,RET_SF
3BC7 75 35 181 LD (IY+40),L
3BC8 74 36 182 LD (IY+41),H
3BC9 76 37 183 LD HL,RET_SF
3BCA 21 02 30 184 LD HL,RET_SF
3BCB 75 38 185 LD (IY+42),L
3BCC 74 39 186 LD (IY+43),H
3BCD 76 3A 187 LD HL,RET_SF
3BCE 21 02 30 188 LD HL,RET_SF
3BCF 75 3B 189 LD (IY+44),L
3BD0 74 3C 190 LD (IY+45),H
3BD1 76 3D 191 LD HL,RET_SF
3BD2 21 02 30 192 LD HL,RET_SF
3BD3 75 3E 193 LD (IY+46),L
3BD4 74 3F 194 LD (IY+47),H
3BD5 76 40 195 LD HL,RET_SF
3BD6 21 02 30 196 LD HL,RET_SF
3BD7 75 41 197 LD (IY+48),L
3BD8 74 42 198 LD (IY+49),H
3BD9 76 43 199 LD HL,RET_SF
3BDA 21 02 30 200 LD HL,RET_SF
3BDB 75 44 201 LD (IY+50),L
3BDC 74 45 202 LD (IY+51),H
3BDD 76 46 203 LD HL,RET_SF
3BDE 21 02 30 204 LD HL,RET_SF
3BDF 75 47 205 LD (IY+52),L
3BE0 74 48 206 LD (IY+53),H
3BE1 76 49 207 LD HL,RET_SF
3BE2 21 02 30 208 LD HL,RET_SF
3BE3 75 4A 209 LD (IY+54),L
3BE4 74 4B 210 LD (IY+55),H
3BE5 76 4C 211 LD HL,RET_SF
3BE6 21 02 30 212 LD HL,RET_SF
3BE7 75 4D 213 LD (IY+56),L
3BE8 74 4E 214 LD (IY+57),H
3BE9 76 4F 215 LD HL,RET_SF
3BEA 21 02 30 216 LD HL,RET_SF
3BED 75 50 217 LD (IY+58),L
3BED 74 51 218 LD (IY+59),H
3BED 76 52 219 LD HL,RET_SF
3BED 21 02 30 220 LD HL,RET_SF
3BED 75 53 221 LD (IY+60),L
3BED 74 54 222 LD (IY+61),H
3BED 76 55 223 LD HL,RET_SF
3BED 21 02 30 224 LD HL,RET_SF
3BED 75 56 225 LD (IY+62),L
3BED 74 57 226 LD (IY+63),H
3BED 76 58 227 LD HL,RET_SF
3BED 21 02 30 228 LD HL,RET_SF
3BED 75 59 229 LD (IY+64),L
3BED 74 5A 230 LD (IY+65),H
3BED 76 5B 231 LD HL,RET_SF
3BED 21 02 30 232 LD HL,RET_SF
3BED 75 5C 233 LD (IY+66),L
3BED 74 5D 234 LD (IY+67),H
3BED 76 5E 235 LD HL,RET_SF
3BED 21 02 30 236 LD HL,RET_SF
3BED 75 5F 237 LD (IY+68),L
3BED 74 60 238 LD (IY+69),H
3BED 76 61 239 LD HL,RET_SF
3BED 21 02 30 240 LD HL,RET_SF
3BED 75 62 241 LD (IY+70),L
3BED 74 63 242 LD (IY+71),H
3BED 76 64 243 LD HL,RET_SF
3BED 21 02 30 244 LD HL,RET_SF
3BED 75 65 245 LD (IY+72),L
3BED 74 66 246 LD (IY+73),H
3BED 76 67 247 LD HL,RET_SF
3BED 21 02 30 248 LD HL,RET_SF
3BED 75 68 249 LD (IY+74),L
3BED 74 69 250 LD (IY+75),H
3BED 76 6A 251 LD HL,RET_SF
3BED 21 02 30 252 LD HL,RET_SF
3BED 75 6B 253 LD (IY+76),L
3BED 74 6C 254 LD (IY+77),H
3BED 76 6D 255 LD HL,RET_SF
3BED 21 02 30 256 LD HL,RET_SF
3BED 75 6E 257 LD (IY+78),L
3BED 74 6F 258 LD (IY+79),H
3BED 76 70 259 LD HL,RET_SF
3BED 21 02 30 260 LD HL,RET_SF
3BED 75 71 261 LD (IY+80),L
3BED 74 72 262 LD (IY+81),H
3BED 76 73 263 LD HL,RET_SF
3BED 21 02 30 264 LD HL,RET_SF
3BED 75 74 265 LD (IY+82),L
3BED 74 75 266 LD (IY+83),H
3BED 76 76 267 LD HL,RET_SF
3BED 21 02 30 268 LD HL,RET_SF
3BED 75 77 269 LD (IY+84),L
3BED 74 78 270 LD (IY+85),H
3BED 76 79 271 LD HL,RET_SF
3BED 21 02 30 272 LD HL,RET_SF
3BED 75 7A 273 LD (IY+86),L
3BED 74 7B 274 LD (IY+87),H
3BED 76 7C 275 LD HL,RET_SF
3BED 21 02 30 276 LD HL,RET_SF
3BED 75 7D 277 LD (IY+88),L
3BED 74 7E 278 LD (IY+89),H
3BED 76 7F 279 LD HL,RET_SF
3BED 21 02 30 280 LD HL,RET_SF
3BED 75 80 281 LD (IY+90),L
3BED 74 81 282 LD (IY+91),H
3BED 76 82 283 LD HL,RET_SF
3BED 21 02 30 284 LD HL,RET_SF
3BED 75 83 285 LD (IY+92),L
3BED 74 84 286 LD (IY+93),H
3BED 76 85 287 LD HL,RET_SF
3BED 21 02 30 288 LD HL,RET_SF
3BED 75 86 289 LD (IY+94),L
3BED 74 87 290 LD (IY+95),H
3BED 76 88 291 LD HL,RET_SF
3BED 21 02 30 292 LD HL,RET_SF
3BED 75 89 293 LD (IY+96),L
3BED 74 8A 294 LD (IY+97),H
3BED 76 8B 295 LD HL,RET_SF
3BED 21 02 30 296 LD HL,RET_SF
3BED 75 8C 297 LD (IY+98),L
3BED 74 8D 298 LD (IY+99),H
3BED 76 8E 299 LD HL,RET_SF
3BED 21 02 30 300 LD HL,RET_SF
3BED 75 8F 301 LD (IY+100),L
3BED 74 90 302 LD (IY+101),H
3BED 76 91 303 LD HL,RET_SF
3BED 21 02 30 304 LD HL,RET_SF
3BED 75 92 305 LD (IY+102),L
3BED 74 93 306 LD (IY+103),H
3BED 76 94 307 LD HL,RET_SF
3BED 21 02 30 308 LD HL,RET_SF
3BED 75 95 309 LD (IY+104),L
3BED 74 96 310 LD (IY+105),H
3BED 76 97 311 LD HL,RET_SF
3BED 21 02 30 312 LD HL,RET_SF
3BED 75 98 313 LD (IY+106),L
3BED 74 99 314 LD (IY+107),H
3BED 76 9A 315 LD HL,RET_SF
3BED 21 02 30 316 LD HL,RET_SF
3BED 75 9B 317 LD (IY+108),L
3BED 74 9C 318 LD (IY+109),H
3BED 76 9D 319 LD HL,RET_SF
3BED 21 02 30 320 LD HL,RET_SF
3BED 75 9E 321 LD (IY+110),L
3BED 74 9F 322 LD (IY+111),H
3BED 76 A0 323 LD HL,RET_SF
3BED 21 02 30 324 LD HL,RET_SF
3BED 75 A1 325 LD (IY+112),L
3BED 74 A2 326 LD (IY+113),H
3BED 76 A3 327 LD HL,RET_SF
3BED 21 02 30 328 LD HL,RET_SF
3BED 75 A4 329 LD (IY+114),L
3BED 74 A5 330 LD (IY+115),H
3BED 76 A6 331 LD HL,RET_SF
3BED 21 02 30 332 LD HL,RET_SF
3BED 75 A7 333 LD (IY+116),L
3BED 74 A8 334 LD (IY+117),H
3BED 76 A9 335 LD HL,RET_SF
3BED 21 02 30 336 LD HL,RET_SF
3BED 75 AA 337 LD (IY+118),L
3BED 74 AB 338 LD (IY+119),H
3BED 76 AC 339 LD HL,RET_SF
3BED 21 02 30 340 LD HL,RET_SF
3BED 75 AD 341 LD (IY+120),L
3BED 74 AE 342 LD (IY+121),H
3BED 76 AF 343 LD HL,RET_SF
3BED 21 02 30 344 LD HL,RET_SF
3BED 75 B0 345 LD (IY+122),L
3BED 74 B1 346 LD (IY+123),H
3BED 76 B2 347 LD HL,RET_SF
3BED 21 02 30 348 LD HL,RET_SF
3BED 75 B3 349 LD (IY+124),L
3BED 74 B4 350 LD (IY+125),H
3BED 76 B5 351 LD HL,RET_SF
3BED 21 02 30 352 LD HL,RET_SF
3BED 75 B6 353 LD (IY+126),L
3BED 74 B7 354 LD (IY+127),H
3BED 76 B8 355 LD HL,RET_SF
3BED 21 02 30 356 LD HL,RET_SF
3BED 75 B9 357 LD (IY+128),L
3BED 74 BA 358 LD (IY+129),H
3BED 76 BB 359 LD HL,RET_SF
3BED 21 02 30 360 LD HL,RET_SF
3BED 75 BC 361 LD (IY+130),L
3BED 74 BD 362 LD (IY+131),H
3BED 76 BE 363 LD HL,RET_SF
3BED 21 02 30 364 LD HL,RET_SF
3BED 75 BD 365 LD (IY+132),L
3BED 74 BE 366 LD (IY+133),H
3BED 76 BF 367 LD HL,RET_SF
3BED 21 02 30 368 LD HL,RET_SF
3BED 75 BE 369 LD (IY+134),L
3BED 74 BF 370 LD (IY+135),H
3BED 76 C0 371 LD HL,RET_SF
3BED 21 02 30 372 LD HL,RET_SF
3BED 75 C1 373 LD (IY+136),L
3BED 74 C2 374 LD (IY+137),H
3BED 76 C3 375 LD HL,RET_SF
3BED 21 02 30 376 LD HL,RET_SF
3BED 75 C4 377 LD (IY+138),L
3BED 74 C5 378 LD (IY+139),H
3BED 76 C6 379 LD HL,RET_SF
3BED 21 02 30 380 LD HL,RET_SF
3BED 75 C7 381 LD (IY+140),L
3BED 74 C8 382 LD (IY+141),H
3BED 76 C9 383 LD HL,RET_SF
3BED 21 02 30 384 LD HL,RET_SF
3BED 75 CA 385 LD (IY+142),L
3BED 74 CB 386 LD (IY+143),H
3BED 76 CC 387 LD HL,RET_SF
3BED 21 02 30 388 LD HL,RET_SF
3BED 75 CD 389 LD (IY+144),L
3BED 74 CE 390 LD (IY+145),H
3BED 76 CD 391 LD HL,RET_SF
3BED 21 02 30 392 LD HL,RET_SF
3BED 75 CE 393 LD (IY+146),L
3BED 74 CF 394 LD (IY+147),H
3BED 76 CE 395 LD HL,RET_SF
3BED 21 02 30 396 LD HL,RET_SF
3BED 75 CF 397 LD (IY+148),L
3BED 74 D0 398 LD (IY+149),H
3BED 76 CF 399 LD HL,RET_SF
3BED 21 02 30 400 LD HL,RET_SF
3BED 75 D1 401 LD (IY+150),L
3BED 74 D2 402 LD (IY+151),H
3BED 76 D0 403 LD HL,RET_SF
3BED 21 02 30 404 LD HL,RET_SF
3BED 75 D3 405 LD (IY+152),L
3BED 74 D4 406 LD (IY+153),H
3BED 76 D1 407 LD HL,RET_SF
3BED 21 02 30 408 LD HL,RET_SF
3BED 75 D5 409 LD (IY+154),L
3BED 74 D6 410 LD (IY+155),H
3BED 76 D2 411 LD HL,RET_SF
3BED 21 02 30 412 LD HL,RET_SF
3BED 75 D7 413 LD (IY+156),L
3BED 74 D8 414 LD (IY+157),H
3BED 76 D3 415 LD HL,RET_SF
3BED 21 02 30 416 LD HL,RET_SF
3BED 75 D9 417 LD (IY+158),L
3BED 74 DA 418 LD (IY+159),H
3BED 76 D4 419 LD HL,RET_SF
3BED 21 02 30 420 LD HL,RET_SF
3BED 75 DB 421 LD (IY+160),L
3BED 74 DC 422 LD (IY+161),H
3BED 76 D5 423 LD HL,RET_SF
3BED 21 02 30 424 LD HL,RET_SF
3BED 75 DD 425 LD (IY+162),L
3BED 74 DE 426 LD (IY+163),H
3BED 76 D6 427 LD HL,RET_SF
3BED 21 02 30 428 LD HL,RET_SF
3BED 75 DE 429 LD (IY+164),L
3BED 74 DF 430 LD (IY+165),H
3BED 76 D7 431 LD HL,RET_SF
3BED 21 02 30 432 LD HL,RET_SF
3BED 75 DF 433 LD (IY+166),L
3BED 74 E0 434 LD (IY+167),H
3BED 76 D8 435 LD HL,RET_SF
3BED 21 02 30 436 LD HL,RET_SF
3BED 75 E1 437 LD (IY+168),L
3BED 74 E2 438 LD (IY+169),H
3BED 76 D9 439 LD HL,RET_SF
3BED 21 02 30 440 LD HL,RET_SF
3BED 75 E3 441 LD (IY+170),L
3BED 74 E4 442 LD (IY+171),H
3BED 76 DA 443 LD HL,RET_SF
3BED 21 02 30 444 LD HL,RET_SF
3BED 75 E5 445 LD (IY+172),L
3BED 74 E6 446 LD (IY+173),H
3BED 76 DB 447 LD HL,RET_SF
3BED 21 02 30 448 LD HL,RET_SF
3BED 75 E7 449 LD (IY+174),L
3BED 74 E8 450 LD (IY+175),H
3BED 76 DC 451 LD HL,RET_SF
3BED 21 02 30 452 LD HL,RET_SF
3BED 75 E9 453 LD (IY+176),L
3BED 74 E9 454 LD (IY+177),H
3BED 76 DD 455 LD HL,RET_SF
3BED 21 02 30 456 LD HL,RET_SF
3BED 75 EA 457 LD (IY+178),L
3BED 74 EA 458 LD (IY+179),H
3BED 76 DE 459 LD HL,RET_SF
3BED 21 02 30 460 LD HL,RET_SF
3BED 75 EB 461 LD (IY+180),L
3BED 74 EB 462 LD (IY+181),H
3BED 76 DF 463 LD HL,RET_SF
3BED 21 02 30 464 LD HL,RET_SF
3BED 75 EC 465 LD (IY+182),L
3BED 74 EC 466 LD (IY+183),H
3BED 76 E0 467 LD HL,RET_SF
3BED 21 02 30 468 LD HL,RET_SF
3BED 75 ED 469 LD (IY+184),L
3BED 74 ED 470 LD (IY+185),H
3BED 76 E1 471 LD HL,RET_SF
3BED 21 02 30 472 LD HL,RET_SF
3BED 75 EE 473 LD (IY+186),L
3BED 74 EE 474 LD (IY+187),H
3BED 76 E2 475 LD HL,RET_SF
3BED 21 02 30 476 LD HL,RET_SF
3BED 75 EF 477 LD (IY+188),L
3BED 74 EF 478 LD (IY+189),H
3BED 76 E3 479 LD HL,RET_SF
3BED 21 02 30 480 LD HL,RET_SF
3BED 75 F0 481 LD (IY+190),L
3BED 74 F0 482 LD (IY+191),H
3BED 76 E4 483 LD HL,RET_SF
3BED 21 02 30 484 LD HL,RET_SF
3BED 75 F1 485 LD (IY+192),L
3BED 74 F1 486 LD (IY+193),H
3BED 76 E5 487 LD HL,RET_SF
3BED 21 02 30 488 LD HL,RET_SF
3BED 75 F2 489 LD (IY+194),L
3BED 74 F2 490 LD (IY+195),H
3BED 76 E6 491 LD HL,RET_SF
3BED 21 02 30 492 LD HL,RET_SF
3BED 75 F3 493 LD (IY+196),L
3BED 74 F3 494 LD (IY+197),H
3BED 76 E7 495 LD HL,RET_SF
3BED 21 02 30 496 LD HL,RET_SF
3BED 75 F4 497 LD (IY+198),L
3BED 74 F4 498 LD (IY+199),H
3BED 76 E8 499 LD HL,RET_SF
3BED 21 02 30 500 LD HL,RET_SF
3BED 75 F5 501 LD (IY+200),L
3BED 74 F5 502 LD (IY+201),H
3BED 76 E9 503 LD HL,RET_SF
3BED 21 02 30 504 LD HL,RET_SF
3BED 75 F6 505 LD (IY+202),L
3BED 74 F6 506 LD (IY+203),H
3BED 76 EA 507 LD HL,RET_SF
3BED 21 02 30 508 LD HL,RET_SF
3BED 75 F7 509 LD (IY+204),L
3BED 74 F7 510 LD (IY+205),H
3BED 76 EB 511 LD HL,RET_SF
3BED 21 02 30 512 LD HL,RET_SF
3BED 75 F8 513 LD (IY+206),L
3BED 74 F8 514 LD (IY+207),H
3BED 76 EC 515 LD HL,RET_SF
3BED 21 02 30 516 LD HL,RET_SF
3BED 75 F9 517 LD (IY+208),L
3BED 74 F9 518 LD (IY+209),H
3BED 76 ED 519 LD HL,RET_SF
3BED 21 02 30 520 LD HL,RET_SF
3BED 75 FA 521 LD (IY+210),L
3BED 74 FA 522 LD (IY+211),H
3BED 76 EE 523 LD HL,RET_SF
3BED 21 02 30 524 LD HL,RET_SF
3BED 75 FB 525 LD (IY+212),L
3BED 74 FB 526 LD (IY+213),H
3BED 76 EF 527 LD HL,RET_SF
3BED 21 02 30 528 LD HL,RET_SF
3BED 75 FC 529 LD (IY+214),L
3BED 74 FC 530 LD (IY+215),H
3BED 76 F0 531 LD HL,RET_SF
3BED 21 02 30 532 LD HL,RET_SF
3BED 75 FD 533 LD (IY+216),L
3BED 74 FD 534 LD (IY+217),H
3BED 76 F1 535 LD HL,RET_SF
3BED 21 02 30 536 LD HL,RET_SF
3BED 75 FE 537 LD (IY+218),L
3BED 74 FE 538 LD (IY+219),H
3BED 76 F2 539 LD HL,RET_SF
3BED 21 02 30 540 LD HL,RET_SF
3BED 75 FF 539 LD (IY+220),L
3BED 74 FF 540 LD (IY+221),H
3BED 76 F3 541 LD HL,RET_SF
3BED 21 02 30 542 LD HL,RET_SF
3BED 75 00 543 LD (IY+222),L
3BED 74 00 544 LD (IY+223),H
3BED 76 F4 545 LD HL,RET_SF
3BED 21 02 30 546 LD HL,RET_SF
3BED 75 01 547 LD (IY+224),L
3BED 74 01 548 LD (IY+225),H
3BED 76 F5 549 LD HL,RET_SF
3BED 21 02 30 550 LD HL,RET_SF
3BED 75 02 549 LD (IY+226),L
3BED 74 02 550 LD (IY+227),H
3BED 76 F6 551 LD HL,RET_SF
3BED 21 02 30 552 LD HL,RET_SF
3BED 75 03 551 LD (IY+228),L
3BED 74 03 552 LD (IY+229),H
3BED 76 F7 553 LD HL,RET_SF
3BED 21 02 30 554 LD HL,RET_SF
3BED 75 04 553 LD (IY+230),L
3BED 74 04 554 LD (IY+231),H
3BED 76 F8 555 LD HL,RET_SF
3BED 21 02 30 556 LD HL,RET_SF
3BED 75 05 555 LD (IY+232),L
3BED 74 05 556 LD (IY+233),H
3BED 76 F9 557 LD HL,RET_SF
3BED 21 02 30 558 LD HL,RET_SF
3BED 75 06 557 LD (IY+234),L
3BED 74 06 558 LD (IY+235),H
3BED 76 FA 559 LD HL,RET_SF
3BED 21 02 30 560 LD HL,RET_SF
3BED 75 07 559 LD (IY+236),L
3BED 74 07 560 LD (IY+237),H
3BED 76 FB 561 LD HL,RET_SF
3BED 21 02 30 562 LD HL,RET_SF
3BED 75 08 561 LD (IY+238),L
3BED 74 08 562 LD (IY+239),H
3BED 76 FC 563 LD HL,RET_SF
3BED 21 02 30 564 LD HL,RET_SF
3BED 75 09 563 LD (IY+240),L
3BED 74 09 564 LD (IY+241),H
3BED 76 FD 565 LD HL,RET_SF
3BED 21 02 30 566 LD HL,RET_SF
3BED 75 0A 565 LD (IY+242),L
3BED 74 0A 566 LD (IY+243),H
3BED 76 FE 567 LD HL,RET_SF
3BED 21 02 30 568 LD HL,RET_SF
3BED 75 0B 567 LD (IY+244),L
3BED 74 0B 568 LD (IY+245),H
3BED 76 FF 569 LD HL,RET_SF
3BED 21 02 30 570 LD HL,RET_SF
3BED 75 0C 569 LD (IY+246),L
3BED 74 0C 570 LD (IY+247),H
3BED 76 00 571 LD HL,RET_SF
3BED 21 02 30 572 LD HL,RET_SF
3BED 75 0D 571 LD (IY+248),L
3BED 74 0D 572 LD (IY+249),H
3BED 76 01 573 LD HL,RET_SF
3BED 21 02 30 574 LD HL,RET_SF
3BED 75 0E 573 LD (IY+250),L
3BED 74 0E 574 LD (IY+251),H
3BED 76 02 575 LD HL,RET_SF
3BED 21 02 30 576 LD HL,RET_SF
3BED 75 0F 575 LD (IY+252),L
3BED 74 0F 576 LD (IY+253),H
3BED 76 03 577 LD HL,RET_SF
3BED 21 02 30 578 LD HL,RET_SF
3BED 75 10 577 LD (IY+254),L
3BED 74 10 578 LD (IY+255),H
3BED 76 04 579 LD HL,RET_SF
3BED 21 02 30 580 LD HL,RET_SF
3BED 75 11 579 LD (IY+256),L
3BED 74 11 580 LD (IY+257),H
3BED 76 05 581 LD HL,RET_SF
3BED 21 02 30 582 LD HL,RET_SF
3BED 75 12 581 LD (IY+258),L
3BED 74 12 582 LD (IY+259),H
3BED 76 06 583 LD HL,RET_SF
3BED 21 02 30 584 LD HL,RET_SF
3BED 75 13 583 LD (IY+260),L
3BED 74 13 584 LD (IY+261),H
3BED 76 07 585 LD HL,RET_SF
3BED 21 02 30 586 LD HL,RET_SF
3BED 75 14 585 LD (IY+262),L
3BED 74 14 586 LD (IY+263),H
3BED 76 08 587 LD HL,RET_SF
3BED 21 02 30 588 LD HL,RET_SF
3BED 75 15 587 LD (IY+264),L
3BED 74 15 588 LD (IY+265),H
3BED 76 09 589 LD HL,RET_SF
3BED 21 02 30 590 LD HL,RET_SF
3BED 75 16 589 LD (IY+266),L
3BED 74 16 590 LD (IY+267),H
3BED 76 0A 591 LD HL,RET_SF
3BED 21 02 30 592 LD HL,RET_SF
3BED 75 17 591 LD (IY+268),L
3BED 74 17 592 LD (IY+269),H
3BED 76 0B 593 LD HL,RET_SF
3BED 21 02 30 594 LD HL,RET_SF
3BED 75 18 593 LD (IY+270),L
3BED 74 18 594 LD (IY+271),H
3BED 76 0C 595 LD HL,RET_SF
3BED 21 02 30 596 LD HL,RET_SF
3BED 75 19 595 LD (IY+272),L
3BED 74 19 596 LD (IY+273),H
3BED 76 0D 597 LD HL,RET_SF
3BED 21 02 30 598 LD HL,RET_SF
3BED 75 1A 597 LD (IY+274),L
3BED 74 1A 598 LD (IY+275),H
3BED 76 0E 599 LD HL,RET_SF
3BED 21 02 30 600 LD HL,RET_SF
3BED 75 1B 599 LD (IY+276),L
3BED 74 1B 600 LD (IY+277),H
3BED 76 0F 601 LD HL,RET_SF
3BED 21 02 30 602 LD HL,RET_SF
3BED 75 1C 601 LD (IY+278),L
3BED 74 1C 602 LD (IY+279),H
3BED 76 10 603 LD HL,RET_SF
3BED 21 02 30 604 LD HL,RET_SF
3BED 75 1D 603 LD (IY+280),L
3BED 74 1D 604 LD (IY+281),H
3BED 76 11 605 LD HL,RET_SF
3BED 21 02 30 606 LD HL,RET_SF
3BED 75 1E 605 LD (IY+282),L
3BED 74 1E 606 LD (IY+283),H
3BED 76 12 607 LD HL,RET_SF
3BED 21 02 30 608 LD HL,RET_SF
3BED 75 1F 607 LD (IY+284),L
3BED 74 1F 608 LD (IY+285),H
3BED 76 13 609 LD HL,RET_SF
3BED 21 02 30 610 LD HL,RET_SF
3BED 75 20 609 LD (IY+286),L
3BED 74 20 610 LD (IY+287),H
3BED 76 14 611 LD HL,RET_SF
3BED 21 02 30 612 LD HL,RET_SF
3BED 75 21 611 LD (IY+288),L
3BED 74 21 612 LD (IY+289),H
3BED 76 15 613 LD HL,RET_SF
3BED 21 02 30 614 LD HL,RET_SF
3BED 75 22 613 LD (IY+290),L
3BED 74 22 614 LD (IY+291),H
3BED 76 16 615 LD HL,RET_SF
3BED 21 02 30 616 LD HL,RET_SF
3BED 75 23 615 LD (IY+292),L
3BED 74 23 616 LD (IY+293),H
3BED 76 17 617 LD HL,RET_SF
3BED 21 02 30 618 LD HL,RET_SF
3BED 75 24 617 LD (IY+294),L
3BED 74 24 618 LD (IY+295),H
3BED 76 18 619 LD HL,RET_SF
3BED 21 02 30 620 LD HL,RET_SF
3BED 75 25 619 LD (IY+296),L
3BED 74 25 620 LD (IY+297),H
3BED 76 19 621 LD HL,RET_SF
3BED 21 02 30 622 LD HL,RET_SF
3BED 75 26 621 LD (IY+298),L
3BED 74 26 622 LD (IY+299),H
3BED 76 
```


制御構造って何だろう

[第3回]

Nakamori Akira

中森 章

制御構造というのはなにやら難しそうな用語ですが、要するにプログラムの流れを表現するものです。C言語には、条件分岐や繰り返しといった処理の手順を記述するために豊富な制御構造が用意されているのです。順を追って見ていきましょう。

始めたらやめられない悪魔のゲーム「シムシティー」をなんとか精神力で封印してこの記事の原稿を書いている中森章です。個人的に「ポピュラス」はいまいちだったのですが、これには当分の間のめり込んでしまいそうな予感がします。

さて、今回のテーマは制御構造です。簡単な表現でいうと選択と繰り返しです。かつて構造化プログラミングの父ダイクストラはアルゴリズムは連接、選択、繰り返しの3つの制御構造で記述できるといいました。裏を返せば、最低限いくつかの制御構造がなければアルゴリズムの記述(=プログラム)はできません。プログラムとはアルゴリズムそのもののものです¹⁾。今回はこのありがたい制御構造について学ぶことにしましょう。

¹⁾ そういえば昔「データ構造+アルゴリズム=プログラミング」という名著があった(編集部注: 今でもある)。

制御構造の種類

プログラムには処理の流れというものがあります。それは、まずこれをして、次にあれをして、その次にどれをしてというような処理を行う順序のことです。処理の流れを考えると、通常はある処理が逐次的に実行されるだけですが、ときにはいくつかの選択肢の中からひとつの処理を選んだり、ある処理を何回か繰り返したりすることも必要になります。たとえば、1から与えられた数までの自然数を考えて、その数が奇数ならば加算し、偶数ならば減算した場合の合計値を求めるプログラムを考えてみましょう。これは、数式で表せば、

$$1-2+3-4+5-6+7-\dots$$

の値を求めることです。このプログラムの処理手順は次のようになります。

- 1) 考える自然数の最大値を変数maxに入力する。
- 2) 合計値を保持する変数をsumとしてそれに0を代入する(初期化)。
- 3) 加算または減算する自然数の値を保持する変数をnumberとしてそれに1を代入する(初期化)。
- 4) 変数numberの値が変数maxの値を越えない限り、5)から6)の処理を繰り返す。
- 5) 変数numberの値を2で割った余りが1(奇数)ならば変数sumの値に変数numberの値を加える。そうでなければ(偶数)変数sumの値から変数numberの値を引く。
- 6) 変数numberの値を1だけ増やす。

- 7) 変数sumの値が求める値になっている。

この処理手順において、5)の処理では加算か減算かという2つの処理からひとつの処理の選択を行い、また、5)から6)の処理が4)の処理の条件のもとで繰り返されています。

このようにちょっとしたプログラムの処理手順を考えただけでも選択や繰り返しという処理が必要になってくるのです。実際、選択や繰り返しという処理なしにはプログラムを書くことはできません。そして、このような選択とか繰り返しといったプログラムの処理の逐次的な流れを変更する仕組みをプログラミング用語で制御構造(control structure)と呼んでいます。

制御構造は大きく次の4種類に分類できます。

- 選択制御構造
- 繰り返し制御構造
- 分岐制御構造
- 割り込み処理

これらの処理について簡単に説明しておきましょう。

選択制御構造とは次の3種類の構造の総称です。すなわち、ある条件があって、その条件に合致したとき(あるいは合致しないとき)のみある処理を行う構造。第2に、ある条件に合致したときある処理を行い、合致しないとき別の処理を行う構造。最後に、いくつかの結果を取り得る条件の評価結果にしたがって、それぞれ異なる処理を行う構造です。これらの選択制御構造は時と場合によって微妙に使い分けられます。ただし、すべてのプログラミング言語がこの選択制御構造をすべて備えているわけではありません(C言語にはすべてある)。

繰り返し制御構造は読んで字のごとく、ある処理を繰り返すための構造です。たいていのプログラミング言語には、変数をループカウンタとして使用する繰り返し回数指定型の繰り返し制御構造と、繰り返し処理を終了するための条件を指定する条件指定型の繰り返し制御構造の2種類が備えられています。また、繰り返しの条件判断の時期もループの先頭で行う場合とループの終わりで行うものがあります。

分岐制御構造とは、それまでの処理の流れと直接関係のない部分に処理を変更する制御構造です。BASICなどでお馴染みのGOTO文はこの分岐制御構造に属します。昔からよく議論されるように、分岐を多用するとプログラムの処理があっちに行ったりこっちに来たりで非常に読みにくくなることがあります。このように処理の流れが複雑になったプログラムは「スパゲッティ」と呼ばれ、

プログラムを読む人からは非常に恐れられています。プログラムを書く人ならば一度は耳にするGOTO文廃止論はこのような理由からきているのです。ただし、現在では、GOTO文がまったく不要というわけではなく、エラー発生時にエラー処理に素早く制御を移すためにGOTO文が有用という認識が一般的です。

割り込み処理とはいった発生するかわからない出来事に対して処理をするための制御構造です。たとえば、実行中のプログラムを停止させたいとき私たちはCTRL+Cキーを押します。このキーが押されたときプログラムは中断するのですがそれがいつ押されるかはプログラム側では知ることができません。このような不意の出来事に対処するための処理が割り込み処理なのです。ただし、プログラムの処理によってはいきなり中断されると困ることがあります。たとえば、プログラムの開始時に変更した画面モードやパレットを元に戻すことが必要になるかもしれません。

さて、このようにいくつかある制御構造ですが、最初のうちはすべてをマスターする必要はありません。経験上、通常のプログラムで必要なのは選択と繰り返しだけです。

かつて、構造化プログラミングの提唱者で知られるダイクストラ (E.W.Dijkstra) はその著書『構造化プログラミング』(邦訳：サイエンス社)の中で、プログラミングは接続 (concatenation)、選択 (selection)、繰り返し (repetition) だけで記述できると述べています²⁾。接続とは通常の逐次的処理のことですから、まさに選択と繰り返しの制御構造こそがプログラミングのすべてだと述べているのです。

実際のところ私自身の書いたプログラムを思い出してみても、割り込みはほとんど使用していませんし、分岐に至ってはほんの1回か2回使用したことがあるだけです。初心者は選択と繰り返しさえマスターしておけば大丈夫でしょう。

整理のために、いくつかの基本的な選択制御構造と繰り返し制御構造の処理を流れ図にして図1に示しておきます。

²⁾正確には、ダイクストラは理解しやすいプログラムを書くためには接続、選択、繰り返しのみを用いるのがよいということ述べている。その根拠はそれだけの制御構造で十分という考えがあるのだろう。

C言語の制御構造

先に述べた接続および4種類の制御構造のすべてをC言語では使用することができます。図2に制御構造とC言語で使用する文(や関数)の対応を示しておきます。ただし、今回は図2のすべての文(や関数)を取り上げるのではなく、初心者に必要な接続および選択と繰り返しの焦点を絞って説明します。分岐と割り込みはまた別の機会に譲ります。

●C言語の接続構造

説明するまでもなく、接続とは複数の文の逐次的処理です。ここでは文とは何かということを明確にしておき

図1 いろいろな制御構造

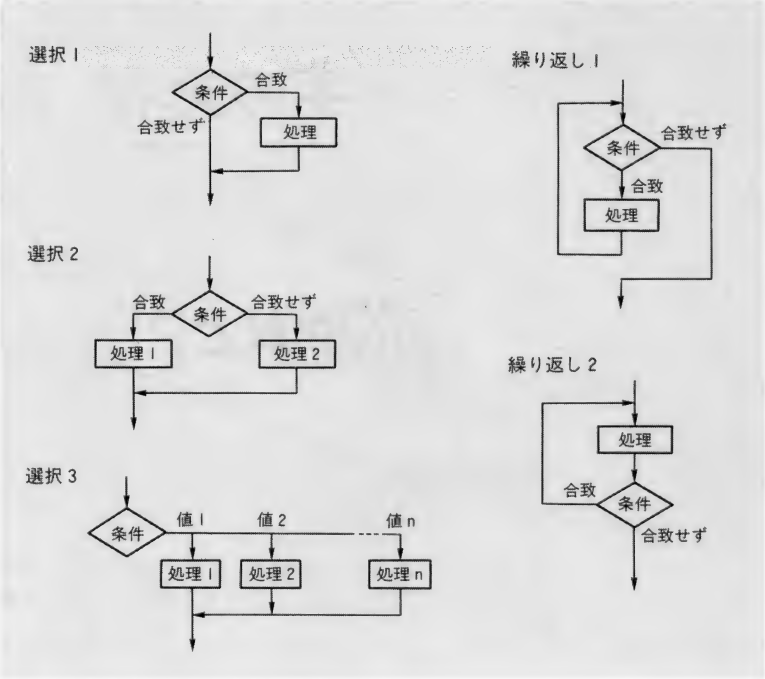


図2 C言語で使用できる制御構造

制御構造	C言語の文(関数)
選択制御構造	if else switch case default
繰り返し制御構造	while for do while
分岐制御構造	goto return setjmp longjmp (break) (continue)
割り込み処理	signal raise

ましょう。C言語における文とは「式のあとにセミコロン(;)を付けたもの」として定義されています。たとえばx=x+1とかprintf(“Hello¥n”)といった式(関数呼び出しも式の種類)はセミコロンを付けて、

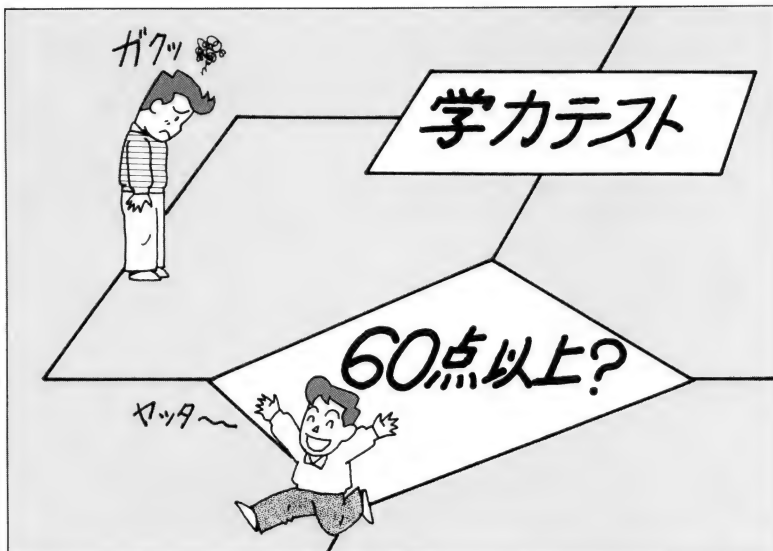
```
x=x+1;
printf(“Hello¥n”);
```

とすることで文として認識されるようになります。注意しなければならないのはセミコロンの扱いです。PASCALではセミコロンは文と文の区切りを表す役割を持てていましたが、C言語ではある式が単なる「式」であるか「文」であるかを区別するための識別子なのです。

C言語ではいくつかの文をひとまとまりにして単一の文と同等に扱うことができます。これが複文またはブロックと呼ばれる構造です。複文は複数の文を波カッコ{ }で囲んだものです。たとえば、

```
{ x=x+1; printf(“Hello¥n”); }
```

が複文です。複文の終わりにはセミコロンは付きません。複文は文法的にひとつの文しか記述できない場所に複数



の文を書くために用いられます。なお、複文は単一の文と同じ扱いですから、複文に含まれる文はそれ自体が複文でもよく、

```
{ x=x+1; { y=x; z=x*x; } x=y+z; }
```

などという文も文法上は可能です（複文にはセミコロンが付かないことに注意しよう）。

ところで、C言語には、

```
;
```

というようにセミコロンだけからなり、何の処理も行わない空文というものがあります。空文は繰り返し制御構造の処理部で積極的に空ループを作る場合に用います。初心者が空文を使用することはあまりないと思いますが、他人のプログラムを読むための基礎知識として覚えておきましょう。同様の考えで、

```
{ }
```

という空ブロックもあります。

●C言語の選択制御構造

C言語の選択制御構造は通常if文とswitch文と呼ばれる構造です。C言語では図1の選択制御構造に示す3種類の構造をすべて備えていて、それぞれは文法的には次のような形式で使用されます。

1) 選択1

- if (exp) stmt;
式expの値が0でないなら³⁾文stmt⁴⁾を実行し、式expの値が0なら何もしない。
(stmt=statement)
- if (exp) { stmt1; stmt2; }
式expの値が0でないなら文stmt1, stmt2,を実行し、式expの値が0なら何もしない。

2) 選択2

- if (exp) stmt1 ; else stmt2 ;
式expの値が0でないなら文stmt1を実行し、式expの値が0なら文stmt2を実行する。
- if (exp) { stmt11; stmt12; }
else stmt21 ;
式expの値が0でないなら文stmt11, stmt12,を実行し、式expの値が0なら文stmt21を

実行する。

- if (exp) stmt1 ;
else { stmt21; stmt22; }
式expの値が0でないなら文stmt1を実行し、式expの値が0なら文stmt21, stmt22.....を実行する。
- if (exp) { stmt11; stmt12; }
else { stmt21; stmt22; }
式expの値が0でないなら文stmt11, stmt12,を実行し、式expの値が0なら文stmt21, stmt22.....を実行する。

3) 選択3

- switch (exp) {
case val1 : stmt11; stmt12;
case val2 : stmt21; stmt22;
:
case valn : stmtn1; stmtn2;
}
式expの値がval1ならばstmt11, stmt12.....を、式expの値がval2ならばstmt21, stmt22.....を、
:
式expの値がvalnならばstmtn1, stmtn2.....を実行する⁵⁾。
- switch (exp) {
case val1 : stmt11; stmt12;
case val2 : stmt21; stmt22;
:
case valn : stmtn1; stmtn2;
default : stmt1; stmt2;
}
式expの値がval1ならばstmt11, stmt12.....を、式expの値がval2ならばstmt21, stmt22.....を、
:
式expの値がvalnならばstmtn1, stmtn2.....を実行するが、式expの値がval1からvalnのどれとも一致しないならstmt1, stmt2.....を実行する。

●C言語の繰り返し制御構造

C言語での繰り返し制御構造にはwhile文、for文、do文があります。最初に文法を説明しましょう。

1) while文

- while (exp) stmt;
式expの値が0でない間、文stmtを繰り返す。
- while (exp) { stmt1; stmt2; }
式expの値が0でない間、文stmt1, stmt2.....を繰り返す。

2) for文

- for (exp1; exp2; exp3) stmt;
まず、式 (文) exp1を実行し、式exp2の値が0でない間、文stmtおよび式 (文) exp3の実行を繰り返す。
- for (exp1; exp2; exp3) { stmt1; stmt2; }
まず、式 (文) exp1を実行し、式exp2の値が0でない間、文stmt1, stmt2.....および式 (文)

exp3の実行を繰り返す。

3) do文

・do stmt; while (exp);

まず文stmtを実行し、式expの値が0でない間、文stmtを繰り返す。

・do { stmt1; stmt2; …… } while (exp);

まず、文stmt1, stmt2, ……を実行し、式expの値が0でない間、文stmt1, stmt2……を繰り返す。

先に繰り返し制御の種類として、繰り返し回数指定型と終了条件指定型があると説明しましたが、基本的にはfor文が繰り返し回数指定型、while文とdo文が終了条件指定型になります。while文とdo文の違いは繰り返し処理を一度も実行しないことがあるか、最低1回は実行するかの違いです。とはいってもこれらのfor文の形式を見てもどこが繰り返し回数指定型なのかかわからないと思います。どう見ても終了条件指定型ですね。これはfor文の使用例を見ないとわかりません。for文は次のような形式で使用されることがほとんどです。

```
for(i=0; i<100; i=i+1) {
    変数 i を使用する処理
}
```

これは、最初に変数 i の値を 0 に初期化(i=0)し、変数 i の値が100より小さい(i<100)間、変数 i を使用する処理 (別に変数 i を使用しなくてもよいが) および変数 i の更新(i=i+1)を繰り返すという記述です。これは繰り返し条件指定型ですね。つまり、for文は応用上は繰り返し回数指定型なのですが、C言語ではもっと一般性を持たせて終了条件指定型にしてあるのです。

C言語の繰り返し制御構造は、繰り返し回数の指定、終了条件の指定といった使い分けらしきものはありますが、結局はどれも終了条件指定型の繰り返しです。このため、すべての制御構造はwhile文で構成することができます。たとえばfor文とdo文はwhile文を用いて次のように変換することもできます。

```
for ( exp1; exp2; exp3 ) stmt;
→ exp1; while ( exp2 ) { stmt; exp3; }
```

```
do stmt; while ( exp );
```

```
→ stmt; while ( exp ) stmt;
```

このように、C言語の繰り返し制御構造は効能としてはどれも似たようなものですから、それぞれの使用頻度は人によってかなり偏っています。たとえば、for文やdo文をまったく使わずにwhile文だけでプログラムを書く人もいます。傾向としては繰り返し回数が決まっているときはfor文を使い、それ以外はwhile文かdo文の一方のみを使用するという人が多いようです。

●必殺技, breakとcontinue

一般に繰り返し制御構造というのはループの先頭または最後で終了条件の判定が行われます。ところがある場合にはループの途中でループを終了したくなったり、以降の処理を無視したくなることがあります。そのための制御構造がC言語には用意されています。それがbreak文とcontinue文です。

break文は、そのbreak文を囲む最小のswitch文か、繰り返し制御構造 (for文, while文, do文) の最小の繰り返し単位の実行を終了させるための文です。break文はその性格上switch文の本体か繰り返し制御構造の本体の中でしか使用することができません。図3にbreak文を使用する場合のswitch文や繰り返し制御構造の制御の流れを示します。

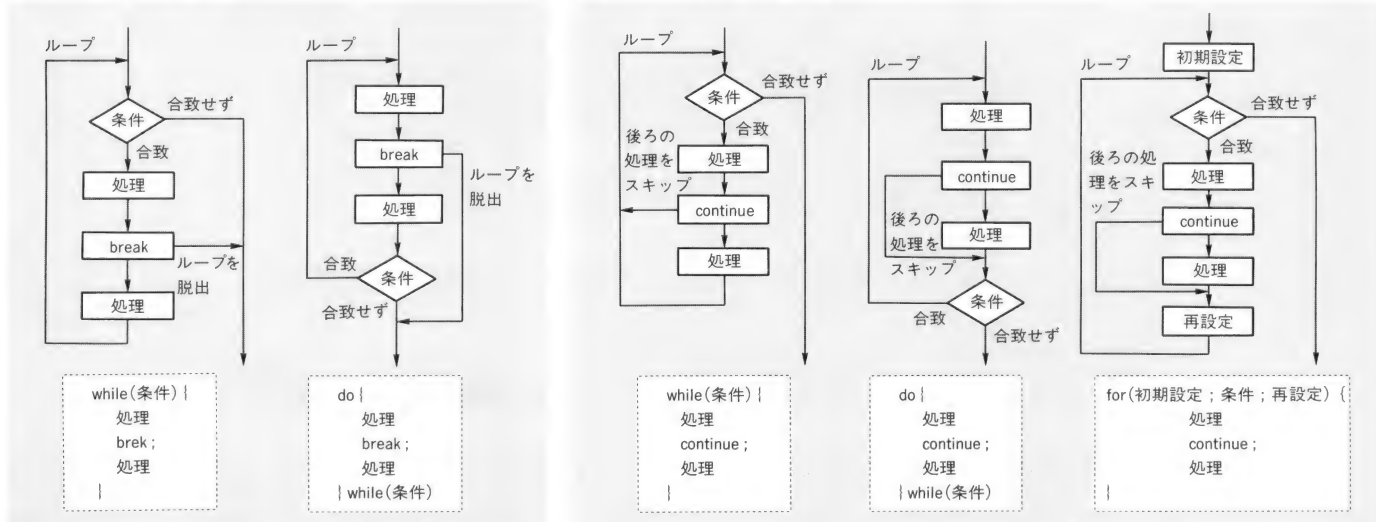
ところで、ここで「最小の」という意味は制御構造が入れ子になっている場合、ひとつだけ外側にのみ抜け出すことができるということです。たとえば、

```
while(i<1000) {
    n=n*i;
    for(j=0; j<2000; j=j+1) {
        if (j>n) break;
        n=n+j;
    }
    i=i*i;
}
```

という入れ子の繰り返し構造を考えましょう (このプログラムはまったく無意味ですから何をやっているのか考えるのはやめましょう)。if文の条件が成立してbreak文

図3 break文の制御の流れ

図4 continue文の制御の流れ



が実行されると、内側のループであるfor文は終了してしましますが、外側のループであるwhile文は影響を受けません。

一方continue文は、continue文を含む最小の繰り返し制御構造のループの終わりまで処理をスキップするための文です。continue文は繰り返し制御構造の中でcontinue文以降の文を無視して次の繰り返しの移りたいときに使用します。continue文はbreak文とは違い、switch文の中に書くことはできません。図4にcontinue文を使用する場合の繰り返し制御構造の制御の流れを示します。

continue文はぜひとも必要な文というわけではありません。これはelse付きのif文や字下げを多用することによってプログラムが見にくくなるのを防ぐために使用します。たとえば、

```
while(i<1000) {
    i=i*i;
    if(i<100) {
        簡単な処理
    }
    else {
        複雑な処理
    }
}
```

リスト1 while文を使用したプログラム

```
1: /*
2:   リスト1   *** while を使ったプログラム ***
3: */
4:
5: main()
6: {
7:     int max,sum,number;
8:
9:     printf("最大値?");
10:    scanf("%d", &max);
11:
12:    sum = 0;
13:    number = 1;
14:    while( number <= max ){
15:        if( (number % 2)==1 ) /* 2で割った余りが1 → 奇数 */
16:            sum = sum+number;
17:        else /* 2で割った余りが0 → 偶数 */
18:            sum = sum-number;
19:        number = number+1;
20:    }
21:
22:    printf("結果 = %d\n", sum);
23: }
```

リスト2 for文を使用したプログラム

```
1: /*
2:   リスト2   *** for を使ったプログラム ***
3: */
4:
5: main()
6: {
7:     int max,sum,number;
8:
9:     printf("最大値?");
10:    scanf("%d", &max);
11:
12:    sum = 0;
13:    for ( number = 1; number <= max; number=number+1 ){
14:        switch ( number % 2 ){
15:            case 1:
16:                sum = sum+number;
17:                break;
18:            case 0:
19:                sum = sum-number;
20:                break;
21:        }
22:    }
23:
24:    printf("結果 = %d\n", sum);
25: }
```

というプログラムがある場合、continue文を使って、

```
while(i<1000) {
    i=i*i;
    if(i<100) {
        簡単な処理
        continue;
    }
    複雑な処理
}
```

と書けば、else文による字下げを省略でき、プログラムがスッキリとします。

³⁾ C言語で条件を示す場合、0が偽を、0以外が真を示す。if文などの条件指定としては、 $x < y$ などという条件式(関係式)が普通であるが、 $x+y$ などの一般の(整数を値とする)式も使用できる。C言語の条件式は、指定した関係が満たされる場合は1、それ以外の場合は0が値となる。

⁴⁾ 正確には文ではなくstmtと書くべきである。if、else、do、whileなどの処理部には単一の文または単一の複文しか記述できないのでセミコロンが付いた文では間違いである(これでは文+空文の2文である)。しかし、この連載では視覚的な理解を第一として、ひとつの文をあえてstmtと表現している。

⁵⁾ 実はこの説明は正しくない(こればっか)。switch文では条件式に一致する値を持つcaseラベル以降のすべてのcaseラベル(defaultを含む)の処理を順次実行する(caseの突き抜け)。このため、条件式の値ごとに処理を切り分けたいのであれば、ひとつのcaseラベルの処理が終了したあとにswitch文から抜け出すための明示的な指定をしなければならない。そのための指定はbreak文またはreturn文で行う。C言語のswitch文がこのような仕様になっている理由は多重のcaseラベルに対して単一の処理を記述できるようにするためである。それ以外の目的での使用は健全でないというK&Rにも書かれている。

◆基礎力を高めよう

設問1 C言語ではいくつかの文を{ }で囲んで複文を作り、PASCALではいくつかの文をbeginとendで囲んで複文を作ります。複文内におけるセミコロンの扱いの違いを例を挙げて説明してください。

設問2 次のif文の解釈として正しいのは、それぞれA)、B)のうちどちらでしょう。

if (E1) if (E2) S1 ; else S2 ;

→

A) if (E1) {
 if (E2) S1 ; else S2 ;
}

※elseはif(E2)に対応する。

B) if (E1) {
 if (E2) S1 ;
} else {
 S2 ;
}

※elseはif(E1)に対応する。

(解答は138ページ)

制御構造を用いたプログラム

C言語の制御構造について主なものをひと通り説明してきました。ここでは学んだ制御構造を使って実際のプログラムを作ってみましょう。例題として、今回の制御

構造の説明の最初に挙げた、

1-2+3-4+5-……

を求めるプログラムをいろいろな制御構造を用いて書いてみましょう。プログラムの形式は先月号で示したテンプレート（定型的な枠組み）にあてはめ、scanf関数で自然数の最大値を入力し、求める値を計算したあと、printf関数で計算結果を出力するものとします。リスト1、リスト2、リスト3が、それぞれwhile文、for文、do文を使用したプログラムです。興味のある人は実行してみてください。

なお、リスト1とリスト3では、加算（または減算）する自然数が偶数であるか奇数であるかの判断はif文で行っていますが、リスト2ではswitch文を使ってみました。ひとつのアルゴリズムを実現するための方法はひとつではないということを実感してください。

ところで、リスト1からリスト3のプログラムはあまり効率がよくありません。

1-2+3-4+5-……

という式を見たとき、

$$\begin{aligned} & 1-2+3-4+\cdots+(2n) \\ &= (1-2) + (3-4) + \cdots + [(2n-1)-(2n)] \\ &= (-1) + (-1) + \cdots + (-1) \\ &= -n \\ & 1-2+3-4+\cdots+(2n+1) \\ &= 1+(-2+3)+(-4+5)+\cdots+[-(2n)+(2n+1)] \\ &= 1+1+\cdots+1 \\ &= n+1 \end{aligned}$$

を発見すればリスト4のようなプログラムを書くこともできます。リスト1からリスト3のプログラムは与える自然数の上限値が大きくなるにつれて計算時間が増えてきますが、リスト4ではどれも同じ時間で計算できてしまいます⁶⁾。

ときには問題をよく吟味してプログラムを書くことも必要ですね。と教訓を得たところで今回はこのくらいにしておきましょう。

⁶⁾天才数学者のガウスは少年時代、1から与えられた数値までの和を即座に答えて彼の先生を驚かせたという。ガウスはすでに等差数列の和の公式を知っていたわけだ。このように本筋とは違う方法で効率のよいプログラムを書くのがガウスのように偉くなったような気分になれるかも。

今月はここまで

今回制御構造を覚えることで逐次的処理しかやれなかったC言語のプログラムにも自由度が増えてきました。これでやっとC言語によるプログラミングの入り口にさしかかったことになります。C言語で本当に「使える」プログラムを作るためにはまだまだ覚えなければならぬことがたくさんありますが、プログラミングの中で今回の制御構造がもっとも大切なものです。各自、制御構造を応用したいろいろなプログラムを書いてみて、使い方を着実に身につけるようにしてください。来月は配列と文字列について説明したいと思います。それではまたお会いしましょう。

◆参考文献

C compiler PRO-68K各マニュアル、シャープ

プログラミング言語C第二版、共立出版

◆基礎力を高めようの解答

設問1 セミコロンはC言語では文の終わり、PASCALでは文と文の区切りを表す。したがって、C言語では（複文でない）文の終わりには必ずセミコロンが必要であるが、PASCALでは文が連続していない場合はセミコロンは必要ない。この違いは複文内の最後の文に現れる。PASCALでは複文内の最後の文、すなわちendの直前の文の最後にセミコロンを付けても付けなくてもよい。しかし、C言語では } の直前の文の最後には（それが複文でなければ）必ずセミコロンが必要である。

すなわち、PASCALでは、

```
begin x:=x+1; y:=y+1; end
```

```
begin x:=x+1; y:=y+1 end
```

はどちらも正しい複文である。

C言語では、

```
{ x=x+1; y=y+1; }
```

は正しいが、

```
{ x=x+1; y=y+1 }
```

は正しくない。

設問2 A)が正しい。elseはそれより前の一番近いifに対応する。同じような考え方で、

```
if (E1) S1; else if (E2) S2; else S3;
```

は、

```
if (E1) S1; else { if (E2) S2; else S3; }
```

であることがわかる。

リスト3 do文を使用したプログラム

```
1: /*
2: リスト3 *** do ~ while を使ったプログラム ***
3: */
4:
5: main()
6: {
7:     int max,sum,number;
8:
9:     printf("最大値?");
10:    scanf("%d", &max);
11:
12:    sum = 0;
13:    number = 1;
14:    do {
15:        if( (number % 2) == 1 ) /* 奇数 */
16:            sum = sum+number;
17:        else /* 偶数 */
18:            sum = sum-number;
19:        number = number+1;
20:    } while( number <= max );
21:
22:    printf("結果 = %d\n", sum);
23:
24: }
```

リスト4 もっとも効率のよいプログラム

```
1: /*
2: リスト4 *** もっとも効率のいいプログラム ***
3: */
4:
5: main()
6: {
7:     int max,sum; /* number の宣言は要らない */
8:
9:     printf("最大値?");
10:    scanf("%d", &max);
11:
12:    switch ( max % 2 ) { /* 普通はif文で書く */
13:    case 1:
14:        sum = max/2 + 1;
15:        break;
16:    case 0:
17:        sum = -max/2;
18:        break;
19:    }
20:
21:    printf("結果 = %d\n", sum);
22: }
```


マシン語カクテル in Z80's Bar

第17回——私はエディタ,原稿まだかな——

シナリオ & イラスト：山田純二

特別監修：浦川博之



前回のややこしい雰囲気を引きずりながら進んでいく今回のマシン語カクテル。さて、プログラムはという山田君がなんの脈絡もなく変な扮装で持ってきてくれるS-OS用のカーソルエディタです。どのようにでも自由にご使用ください。

♪ カラン、コローン

マスター (以下M)：いらっしやい。

メアリー (以下メ)：イラッシャーイ。

長老 (以下老)：どうも、ひさしぶりじゃな。あい変わらず静かじゃのう。

M：余計なお世話ですよ、今日はおひとりで？

老：うむ、2,3人若い連中をつれてこようと思ったが、みな忙しそうだったからの。

M：そういえば、常連の山田君も最近顔を見せないな。ああそうだ。ようこちゃん、光君は？

ようこ (以下Yo)：知らないわよ。

老：冷たいもんじゃのう。ところで、ようこちゃん、その手に持ってるものはなにかな。

Yo：この間、マスターが骨董品屋で買ってきた壺よ。

老：ずいぶんとホコリにまみれているし、形も妙な壺じゃな。

M：なんでも、インドのとある寺院に置いてあったものだということですよ。

老：ふーん、本当かのう。

Yo：いいじゃない。どうせ安かったんでしょ。

メ：そういうコトデース。汚れたママジャカワイソウだから、キレイにシテアゲマシヨウ (キュッキュツ)。

老：げほっげほつ。ものすごいホコリじゃ……。ヘックショーイっとな。

壺：モクモクモク〜。

M：うわっ、なんだこの煙は。ようこちゃん、消火器！ 消火器！

？：ハッハッハ〜、ダイジョウブイ、じゃなくて、呼ばれて飛び出てじゃじゃじゃじゃ〜ん。

Yo：あっ、謎のアラビア人。

メ：Oh！ 違イマ〜ス。ハクション大魔王デース。

老：なにをいっておるんじゃ。なんだ、誰かと思えば山田君じゃないか。凝った変装をしておて。

純二 (以下純)：そのわりにはあっさりばれてしまいましたね。

M：今日はなにかプログラムを持ってきたんですか。

純：よくぞ聞いてくれました。構想2時間、プログラミングに8時間かけた、テキストエディタを持ってきたんですよ。



エディタとは？

老：ところで、ようこちゃんエディタとはなにか知っているかの。

Yo：知っているわよ。雑誌なんかでライターと呼ばれる人たちを使って原稿を集め、きびしい進行に泣きながら、本を作っている人たちのことでしょう。

純：それは、編集者。

老：たしかに編集者のこともエディタというが、この場合は……。

メ：コンピュータのプログラムヲ書クタメノ道具デスネ。

老：メアリーのいうとおりじゃ。プログラミングをするときに、作成したいプログラムを管理するプログラムのことをいっているのじゃろう。

Yo：プログラムを作るためのプログラム。いまいちピンとこないなあ。

純：たいていの場合、そんなことは気にしないで使っているからね。

M：ところで純二君、今回はどういったタイプのエディタを作ってきたんですか。

純：BASICやZEDAに付属しているカーソルエディタです。

メ：ソレハ、ドウィッタモノデスカ。

老：1行ごとについている行番号によって、それぞれの行をエディットしていくものじ

やよ。

Yo：ほかにはどういった種類があるの。

老：ラインエディタ、スクリーンエディタなんかがあるのう。

Yo：いちばん使いやすいものはどれなの。

老：やっぱり、編集するものの中を自由に行ったり来たりして、どこでも編集できるスクリーンエディタじゃな。

Yo：ふーん、だったら純二君、あまり使い勝手のよくないカーソルエディタじゃなくてスクリーンエディタを作ればよかったのに。

純：たしかにそうかもしれませんが、スクリーンエディタはカーソルエディタに比べてはるかに複雑なんですよ。スクリーンエディタを作るとなると、かなりの行数が必要となるけど、カーソルエディタならほんの数百行ですみますから。

メ：ソウソウ、小サイコトハいいコトネ。

M：なんのこっちゃ。



さあ、作るぞ

老：さて、ようこちゃん、編集という作業はどういうものかわかるかな？

Yo：文字を入力すること。

老：ちょっと違うな、編集というのは入力されたものを〇〇、☆☆したり？？したりすることをいうんじやよ。

Yo：やっだ〜、長老のス・ケ・ベ。

老：な、なにをを考えておるのじゃ。冗談いってないで真面目に答えなさい。

純：長老のいけず〜う。

老：ばかもの(ポカッ)！ おぬしまで一緒になってなにをいっておるんじや。

純：痛いなあ……。わざわざ伏せ字にしていうから変な誤解をまねくんですよ。答えは、削除、追加、更新することでしょう。

老：そういうことじゃ。素直に答えておれ

ば痛い目に遭わずにすんだものを。

メ：冗談ハソレグライニシテ、具体的ナ方法ヲ、教エテクダサイ。

老：よかろう。まず、テキストがどのような構成となっているかを説明してくれんか、純二君。

純：はい。S-OSの場合、行のエンドコードは0D_Hで、テキストのエンドコードは00_Hとなっています。

老：うむ、次に必要なワークはなにがあるかな。

純：テキストを格納している先頭アドレス(#TEXTENT)、テキストの終了アドレス(#TEXTEND)、編集している行の先頭アドレス(#EDADR)の3つです。

Yo：たった3つだけでいいの。

老：そうじゃよ、なんとなく難しそうに思えてしまうが、やっていることは思ったより単純なんじゃよ。では、行の削除はどうやっているか、見ていくとするか。

純：行の削除は、編集している行の先頭アドレスへ次の行の先頭アドレスから最終行までのテキストを転送してやれば完了です(図1)。

メ：プログラムではドウナッテイルンデスカ？

純：それは、

1) 削除したい行の先頭アドレスをDEレジスタに格納する

2) 削除したい次の行の先頭アドレスをHLレジスタに格納する

3) (#TEXTEND) からHLの値を引いたもの+1をBCレジスタに格納する

4) LDIR命令で転送する

5) DE-1の値を新しい(#TEXTEND)の値として格納する

と、いうふうにやるんですよ。

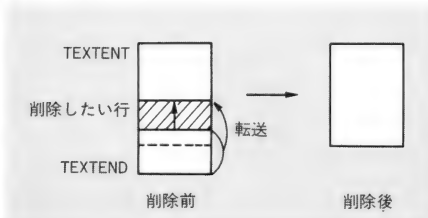
老：次は、行の追加じゃな。

純：行の追加は、追加したい行から最終行までのテキストを、追加する行の桁数だけ下に転送してから、追加する行の内容をメモリに転送してやればいいのです(図2)。具体的な手順は、

1) 追加したい行の先頭アドレスをHLレジスタに求める

2) PUSH HLで値を保存

図1



3) 入力された行の桁数をBCレジスタに求める

4) (#TEXTEND)の値をHLレジスタにロードする

5) HLレジスタにさっき求めたBCレジスタの値を足したものを、DEレジスタの値とする(これが、新しい(#TEXTEND)の値)

6) LDDR命令で転送する

7) POP HLで保存した値を取り出し、そのアドレスに入力された行を転送するという具合になります。

Yo：よし、次は行の更新ね。

純：行の更新には2通りの方法があるんです。まずひとつ目は、更新したい行を削除してから、新しく入力された行を追加する方法。

メ：サッキノ削除ト追加ノプログラムをツカエバイイノデスネ。

純：そのとおり。使えるものはできるかぎり共用してしまおうという精神に基づくものです。プログラムも短く収まるので、とってもうれしいのですが、実行速度の点でもうひとつの方法に劣るところがあるんですよ。

Yo：で、その方法って？

善司(以下善)：あるわけないだよ〜ん。

Yo：純二く〜ん。嘘をついたわね。

純：うわあ、ようこさんマジになって怒らないでくださいよ。それに善ちゃん、いきなりわいて出てきてとんでもないことをいわないでよ。

善：いやあ、みんなで真面目な顔をしているから、場をなごませようとしただけです。

老：いらんことばっかりに、気が回る男じやのう。ま、こんなやつはほっといてさっさと説明してくれんか。

純：はい、削除してから追加するとたしかにプログラムの構造は単純となります。もうひとつの方法とは、更新する行と入力された行の長さを比べて、

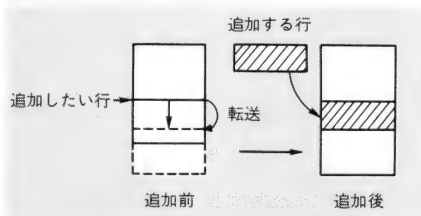
1) 更新する行=入力された行

2) 更新する行>入力された行

3) 更新する行<入力された行

の3通りに場合分けをしてやるのです。1)の場合にはそのまま入力された行を転送し

図2



てやり、2)の場合には余った桁を詰め、3)の場合にははみ出した桁の分空けてやればいいのです。この方法だとブロック転送の回数が1回ですむから効率はいいんですよ。まあ、今回は行数をできるだけ短くしたかったので前者の方法を使いましたけどね。



ZED-645の使用法

M：それじゃあ、プログラムの使い方を説明してくれませんか。

純：使えるコマンドはカコミにあるとおりです。使い方についてはZEDAの付属のエディタとまったく同じです。違うのは行の指定に16進4桁の数値を使っている点でしょう。行数の関係上、10進16進変換ルーチンを入れるスペースがなくなってしまったのです。

Yo：コマンドを見てみると削除と追加のコマンドはあるけど、行の更新はどうやっているの。

純：それは、リストの最初のほうを見てください。1行入力が終わったあと、入力された行の先頭がプロンプト“>”であった場合にはコマンド判定ルーチンへ行き、そうでなかった場合には行の更新を行うのです。“T”コマンドでリストを表示してから変更したい行をエディットしていけばいいんですよ。

老：なるほどな、しかし短いだけあって本当に最低限の機能しかついておらん。

純：まあ、短いというだけあって小回りが利くし、もし、ほしい機能があったなら自分でどんどん拡張していけばいいと思いま

ZED-645コマンド一覧表

nは16進4桁。

Tn……n行からテキストを表示していきます。nを省略した場合は0行目から表示します。

In……n行目からテキストを挿入していきます。このコマンドを実行後はテキスト挿入モードになりますので、抜きたいときにはSHIFT+BREAKを押してください。nを省略した場合には(#EDADR)の指すアドレスからテキストを挿入していきます。

Dn……n行目のテキストを削除します。nは省略不可能です。

Xaddress……テキスト格納アドレスを16進4桁で指定します。アドレスを省略すると現在のメモリの格納状態が表示されます。

&……テキストをクリアします。

R……テキストを復活させます。

Sfile name……現在編集集中のテキストをセーブします。

Lfile name……指定されたファイル名のテキストをロードします。

Q……エディタを終了してS-OSのシステムに戻ります。



すけどね。それが、このエディタの正しい
使い方だと思います。



帰ろ、帰ろ

M: といったところで、そろそろおひらき
ですが、山田君いきなり登場したりして、
今月も光君がやるはずじゃなかったの。

純: 彼もなにかと忙しいようで。

メ: ワタシ、サミシイデス。

老: ようこちゃんも、さみしいじやろう。

Yo: 別に。

善: ま、そういうことにしましょうか。

純: それじゃあ、僕はこれで失礼します。

長老と善ちゃん一緒に帰らない?

老: わしはもう少しここにいることにする
から、先に帰っていいぞ。

善: 僕も、もうちょっと残っていくから。

純: ふーん。じゃあ、またね。

M: ありがとうございます。

♪ カラン、コロ〜ン

つづく

リスト1

```
0000 1 ; ZED-645
0000 2 ; in Z80's Bar
0000 3 ; 1990.10.15 by J.YAMADA
0000 4
0000 5 ORG $A000
0000 6
1FF4 P 7 #PRINT EQU $1FF4
1FF1 P 8 #PRINTS EQU $1FF1
1FEE P 9 #LTNL EQU $1FEE
1FE5 P 10 #MSX EQU $1FE5
1FE8 P 11 #MSG EQU $1FE8
1FE2 P 12 #MPRINT EQU $1FE2
1FC4 P 13 #BELL EQU $1FC4
1FC1 P 14 #PRTHX EQU $1FC1
1FBE P 15 #PRTHL EQU $1FBE
1FB2 P 16 #HLHEX EQU $1FB2
1FC7 P 17 #PAUSE EQU $1FC7
1FD3 P 18 #GETL EQU $1FD3
1F76 P 19 #KBFAD EQU $1F76
2018 P 20 #CSR EQU $2018
201E P 21 #LOC EQU $201E
1FAF P 22 #WOPEN EQU $1FAF
1FAC P 23 #WRD EQU $1FAC
1FA6 P 24 #RDD EQU $1FA6
1FA3 P 25 #FILE EQU $1FA3
1FA0 P 26 #FSAME EQU $1FA0
1FD0 P 27 #FPRNT EQU $1FD0
2009 P 28 #ROPEN EQU $2009
1F72 P 29 #SIZE EQU $1F72
1F70 P 30 #DTADR EQU $1F70
1F6E P 31 #EXADR EQU $1F6E
2033 P 32 #ERROR EQU $2033
1F6A P 33 #MEMAX EQU $1F6A
1F9D P 34 #FPRINT EQU $1F9D
A000 35
A000 18 09 36 COLD JR COLD2
A002 18 1D 37 HOT JR HOT2
A004 38
A004 00 4E 39 #TEXTENT DW $4E00
A006 00 4E 40 #TEXTEND DW $4E00
A008 00 00 41 #EDADR DW 0000
A00A 00 42 #BACKCHR DB 00
A00B 43
A00B 44 COLD2
A00B CD E2 1F 45 CALL #MPRINT
A00E 0C 2A 2A 20 46 DB $0C,"** ZED-645 **",$0D,00
A012 5A 45 44 2D
A016 36 34 35 20
A01A 2A 2A 0D 00
A01E CD 73 A0
A021 48
A021 2A 04 A0 49 LD HL,($TEXTENT)
A024 22 08 A0 50 LD (#EDADR),HL
A027 CD 5A A2 51 CALL #ENDSEARCH
A02A 52
A02A 53 ;COMMAND LINE
A02A 54
A02A 55 COM
A02A 3E 3E 56 LD A,">"
A02C CD F4 1F 57 CALL #PRINT
A02F ED 5B 76 1F 58 LD DE,($KBFAD)
A033 CD D3 1F 59 CALL #GETL
A036 1A 60 LD A,(DE)
A037 21 02 A0 61 LD HL,HOT
A03A E5 62 PUSH HL
A03B FE 3E 63 CP ">"
A03D C2 48 A1 64 JP NZ,#REWRITE
A040 ED 5B 76 1F 65 LD DE,($KBFAD)
A044 13 66 INC DE
A045 1A 67 LD A,(DE)
A046 B7 68 OR A
A047 C8 69 RET
A048 FE 51 70 CP "Q"
A04A 28 25 71 JR Z,EXIT
A04C FE 26 72 CP "E"
A04E 28 23 73 JR Z,#TEXTNEW
A050 FE 54 74 CP "T"
A052 28 59 75 JR Z,#LISTPRT
A054 FE 52 76 CP "R"
A056 28 28 77 JR Z,#RECOVER
A058 FE 44 78 CP "D"
A05A CA E1 A0 79 JP Z,#DELETE
A05D FE 58 80 CP "X"
A05F 28 2A 81 JP Z,#ADRSET
A061 FE 49 82 CP "I"
A063 CA FF A0 83 JP Z,#INSERT
A066 FE 53 84 CP "S"
A068 CA 13 A2 85 JP Z,#SAVE
A06B FE 4C 86 CP "L"
A06D CA BE A1 87 JP Z,#LOAD
A070 C9 88 RET
A071 89 EXIT
A071 E1 90 POP HL
A072 C9 91 RET
A073 92
A073 93 ;TEXT CLEAR
A073 94
```

```
A073 95 #TEXTNEW
A073 2A 04 A0 96 LD HL,($TEXTENT)
A076 7E 97 LD A,(HL)
A077 32 0A A0 98 LD ($BACKCHR),A
A07A 36 00 99 LD (HL),00
A07C 22 06 A0 100 LD ($TEXTEND),HL
A07F C9 101 RET
A080 102
A080 103 ;TEXT RECOVER
A080 104
A080 105 #RECOVER
A080 3A 0A A0 106 LD A,($BACKCHR)
A083 2A 04 A0 107 LD HL,($TEXTENT)
A086 77 108 LD (HL),A
A087 CD 5A A2 109 CALL #ENDSEARCH
A08A C9 110 RET
A08B 111
A08B 112 ;TEXT KAKUNOU ADDRESS
A08B 113
A08B 114 #ADRSET
A08B 13 115 INC DE
A08C CD B2 1F 116 CALL #HLHEX
A08F 30 15 117 JR NC,ADS2
A091 2A 04 A0 118 LD HL,($TEXTENT)
A094 CD BE 1F 119 CALL #PRTHL
A097 3E 2D 120 LD A,"-"
A099 CD F4 1F 121 CALL #PRINT
A09C 2A 06 A0 122 LD HL,($TEXTEND)
A09F CD BE 1F 123 CALL #PRTHL
A0A2 CD EE 1F 124 CALL #LTNL
A0A5 C9 125 RET
A0A6 126 ADS2
A0A6 22 04 A0 127 LD ($TEXTENT),HL
A0A9 CD 73 A0 128 CALL #TEXTNEW
A0AC C9 129 RET
A0AD 130
A0AD 131 ;LIST PRINT
A0AD 132
A0AD 133 #LISTPRT
A0AD 13 134 INC DE
A0AE CD B2 1F 135 CALL #HLHEX
A0B1 38 09 136 JR C,LPRT5
A0B3 5D 137 LD E,L
A0B4 54 138 LD D,H
A0B5 D5 139 PUSH DE
A0B6 CD 76 A1 140 CALL #EDADRSET
A0B9 D1 141 POP DE
A0BA 18 06 142 JR LPRT4
A0BC 2A 04 A0 143 LD HL,($TEXTENT)
A0BF 11 00 00 144 LD DE,0000
A0C2 145
A0C2 7E 146 LPRT4
A0C3 B7 147 LD A,(HL)
A0C4 C8 148 OR A
A0C5 149 RET
A0C5 150 LPRT2
A0C5 EB 151 EX DE,HL
A0C6 CD BE 1F 152 CALL #PRTHL
A0C9 3E 3A 153 LD A,"."
A0CB CD F4 1F 154 CALL #PRINT
A0CE CD E8 1F 155 CALL #MSG
A0D1 CD EE 1F 156 CALL #LTNL
A0D4 CD C7 1F 157 CALL #PAUSE
A0D7 E0 A0 158 DW LPRT3
A0D9 EB 159 EX DE,HL
A0DA 13 160 INC DE
A0DB CD B0 A1 161 CALL #INCADR
A0DE 20 E5 162 JR NZ,LPRT2
A0E0 C9 163 LPRT3
A0E1 164
A0E1 165 ;TEXT 1 LINE DELETE
A0E1 166
A0E1 167 #DELETE
A0E1 13 168 INC DE
A0E2 CD B2 1F 169 CALL #HLHEX
A0E5 38 14 170 JR C,DELERR
A0E7 CD 76 A1 171 CALL #EDADRSET
A0EA 38 0F 172 JR C,DELERR
A0EC 173 #DELSUB
A0EC 5D 174 LD E,L
A0ED 54 175 LD D,H
A0EE CD B0 A1 176 CALL #INCADR
A0F1 CD 8A A1 177 CALL #SIZESET
A0F4 ED B0 178 LDIR
A0F6 1B 179 DEC DE
A0F7 22 06 A0 180 LD ($TEXTEND),HL
A0FA C9 181 RET
A0FB 182 DELERR
A0FB CD C4 1F 183 CALL #BELL
A0FE C9 184 RET
A0FF 185
A0FF 186 ;TEXT INSERT
A0FF 187
A0FF 188 #INSERT
A0FF 13 189 INC DE
A100 CD B2 1F 190 CALL #HLHEX
A103 30 05 191 JR NC,IS2
```



```

A105 2A 08 A0 192 LD HL, (#EDADR)
A108 18 03 193 JR IS4
A10A 194 IS2 CALL #EDADRSET
A10D 195 IS4 LD DE, (#KBPAD)
A10D ED 5B 76 1F 197 CALL #GETL
A111 CD D3 1F 198 LD A, (DE)
A114 1A 199 CP $1B
A115 FE 1B 200 RET Z
A117 C8 201 CALL #INSSUB
A118 CD 2D A1 202 LD DE, (#KBPAD)
A11B ED 5B 76 1F 203 CALL #TRNS
A11F CD A4 A1 204 LD HL, (#EDADR)
A122 2A 08 A0 205 CALL #INCAADR
A125 CD B0 A1 206 LD (#EDADR), HL
A128 22 08 A0 207 JR IS4
A12B 18 E0 208 #INSSUB
A12D 209 PUSH HL
A12D E5 210 CALL #SIZESET
A12E CD 8A A1 211 PUSH BC
A131 C5 212 EX DE, HL
A132 EB 213 CALL #LENGTH
A133 CD 98 A1 214 LD B, 00
A136 06 00 215 LD HL, (#TEXTEND)
A138 2A 06 A0 216 LD E, L
A13B 5D 217 LD D, H
A13C 54 218 EX DE, HL
A13D EB 219 ADD HL, BC
A13E 09 220 LD (#TEXTEND), HL
A13F 22 06 A0 221 EX DE, HL
A142 EB 222 POP BC
A143 C1 223 LDDR HL
A144 ED B8 224 POP HL
A146 E1 225 RET
A147 C9 226 ;TEXT REWRITE
A148 227 #REWRITE
A148 228 CALL #HLHEX
A148 229 RET
A148 230 INC DE
A148 CD B2 1F 231 PUSH DE
A14B D8 232 CALL #EDADRSET
A14C 13 233 POP DE
A14D 05 234 PUSH DE
A14E CD 76 A1 235 POP DE
A151 D1 236 PUSH DE
A152 D5 237 PUSH DE
A153 D5 238 PUSH HL
A154 E5 239 CALL #DELSUB
A155 CD EC A0 240 POP HL
A158 E1 241 POP DE
A159 D1 242 CALL #INSSUB
A15A CD 2D A1 243 POP DE
A15D D1 244 CALL #TRNS
A15E CD A4 A1 245 CALL #CSR
A161 CD 18 20 246 LD L, 05
A164 2E 05 247 CALL #LOC
A166 CD 1E 20 248 LD DE, (#KBPAD)
A169 ED 5B 76 1F 249 CALL #GETL
A16D CD D3 1F 250 LD A, (DE)
A170 1A 251 CP $1B
A171 FE 1B 252 RET Z
A173 C8 253 JR #REWRITE
A174 18 D2 254 ; EDIT POINTER SET
A176 255 ; IN HL=EDIT LINE NO.
A176 256 #EDADRSET
A176 257 EX DE, HL
A176 258 LD HL, (#TEXTEND)
A176 259 EDS2
A176 EB 260 LD A, D
A177 2A 04 A0 261 OR E
A17A 7A 262 JR Z, EDS3
A17B B3 263 CALL #INCAADR
A17C 28 08 264 JR C, EDS3
A17E CD B0 A1 265 DEC DE
A181 38 03 266 JR EDS2
A183 1B 267 EDS3
A184 18 F4 268 LD (#EDADR), HL
A186 270 RET
A186 22 08 A0 271 ; (TEXTEND)-HL
A189 C9 272 #SIZESET
A18A 273 PUSH DE
A18A D5 274 EX DE, HL
A18B EB 275 LD HL, (#TEXTEND)
A18C 2A 06 A0 276 OR A
A18F B7 277 SBC HL, DE
A190 ED 52 278 LD C, L
A192 4D 279 LD B, H
A193 44 280 INC BC
A194 03 281 EX DE, HL
A195 EB 282 POP DE
A196 D1 283 RET
A197 C9 284 ;1 LINE LENGTH
A198 285 #LENGTH
A198 286 LD C, 00
A198 287 LEN2
A198 0E 00 288 INC C
A19A 294 LD A, (HL)
A19B 7E 295 OR A
A19C B7 296 RET Z
A19D C8 297 CP $0D
A19E FE 0D 298 RET Z
A1A0 C8 299 INC HL
A1A1 23 300 JR LEN2
A1A2 18 F6 301 ;BAFA TO MEMORY
A1A4 302 ; IN DE=BAFA ADDRESS, HL=KAKUNOU ADDRESS
A1A4 303 #TRNS
A1A4 304 LD A, (DE)
A1A4 305 OR A
A1A5 B7 306 JR Z, TRS2
A1A6 28 05 307 LD (HL), A
A1A8 77 308 INC HL
A1A9 23 309 INC DE
A1AA 13 310 JR #TRNS
A1AB 18 F7 311 TRS2
A1AD 312 LD (HL), $0D
A1AD 36 0D 313 RET
A1AF C9 316

```

```

A1B0 317 ;1 LINE ADDRESS FORWARD
A1B0 318 #INCAADR
A1B0 319 LD A, (HL)
A1B0 320 OR A, #INCA4 ;TEXT END
A1B0 7E 321 JR Z, #INCA4
A1B1 B7 322 INC HL
A1B2 28 08 323 CP $0D
A1B4 23 324 JR NZ, #INCAADR
A1B5 FE 0D 325 LD A, (HL)
A1B7 20 F7 326 OR A
A1B9 7E 327 RET NZ
A1BA B7 328 #INCA4
A1BB C0 329 SCF
A1BC 330 RET
A1BC 37 331 ;TEXT LOAD
A1BD C9 332 #LOAD
A1BE 333 INC DE
A1BE 334 LD A, 04
A1BE 13 335 CALL #FILE
A1BF 3E 04 336 #LOAD3
A1C1 CD A3 1F 337 CALL #ROPEN
A1C4 340 JR C, LOADERR
A1C4 CD 09 20 341 JR Z, #LOAD4
A1C7 38 3C 342 CALL #FPRINT
A1C9 28 08 343 CALL #LTNL
A1CB CD 9D 1F 344 JR #LOAD3
A1CE CD EE 1F 345 #LOAD4
A1D1 18 F1 346 LD HL, (#TEXTEND)
A1D3 347 LD HL, (#TEXTEND)
A1D3 2A 06 A0 348 PUSH HL
A1D6 E5 349 DEC HL
A1D7 2B 350 CALL #UCHK1
A1D8 CD 4E A2 351 POP HL
A1DB E1 352 JR NZ, LOAD2
A1DC 20 01 353 POP HL
A1DE 2B 354 DEC HL
A1DF 355 LOAD2
A1DF 22 70 1F 356 LD (#DTADR), HL
A1E2 ED 5B 72 1F 357 LD DE, (#SIZE)
A1E6 19 358 ADD HL, DE
A1E7 38 20 359 C, LOADERR2
A1E9 ED 5B 6A 1F 360 LD DE, (#MEMAX)
A1ED B7 361 OR A
A1EE ED 52 362 SBC HL, DE
A1F0 30 17 363 JR NC, LOADERR2
A1F2 11 68 A2 364 LD DE, MMSTR3
A1F5 CD E5 1F 365 CALL #MSX
A1F8 CD 9D 1F 366 CALL #FPRINT
A1FB CD A6 1F 367 CALL #RDD
A1FE CD EE 1F 368 CALL #LTNL
A201 CD 5A A2 369 CALL #ENDSEARCH
A204 D0 370 RET NC
A205 371 LOADERR
A205 CD 33 20 372 CALL #ERROR
A208 C9 373 RET
A209 374 LOADERR2
A209 11 79 A2 375 LD DE, MMSTR5
A20C CD E5 1F 376 CALL #MSX
A20F CD EE 1F 377 CALL #LTNL
A212 C9 378 RET
A213 379 ;TEXT SAVE
A213 380 #SAVE
A213 381 INC DE
A213 13 382 LD A, 04
A214 3E 04 383 CALL #FILE
A216 CD A3 1F 384 LD HL, 0000
A219 21 00 00 385 LD (#DTADR), HL
A21C 22 70 1F 386 LD (#EXADR), HL
A21F 22 6E 1F 387 LD HL, (#TEXTEND)
A222 2A 06 A0 388 LD DE, (#TEXTEND)
A225 ED 5B 04 A0 389 PUSH DE
A229 D5 390 OR A
A22A B7 391 SBC HL, DE
A22B ED 52 392 INC HL
A22D 23 393 LD (#SIZE), HL
A22E 22 72 1F 394 LD DE, MMSTR4
A231 11 71 A2 395 CALL #MSX
A234 CD E5 1F 396 CALL #FPRINT
A237 CD 9D 1F 397 CALL #LTNL
A23A CD EE 1F 398 CALL #WOPEN
A23D CD AF 1F 400 JR C, SERR
A240 38 08 401 POP HL
A242 E1 402 LD (#DTADR), HL
A243 22 70 1F 403 CALL #WRD
A246 CD AC 1F 404 RET NC
A249 D0 405 SERR
A24A 406 CALL #ERROR
A24A CD 33 20 407 LD DE, MMSTR4
A24D C9 408 RET
A24E 409 ;ADDRESS CHECK
A24E 410 #UCHK1
A24E 411 LD DE, (#TEXTEND)
A24E ED 5B 04 A0 412 OR A
A252 B7 413 SBC HL, DE
A253 ED 52 414 RET Z
A255 C8 415 ADD HL, DE
A256 19 416 OR 1
A257 F6 01 417 RET
A259 C9 418 ;TEXT END ADDRESS CHECK
A25A 419 #ENDSEARCH
A25A 420 SUB A
A25A 97 421 LD BC, 0000
A25B 01 00 00 422 LD HL, (#TEXTEND)
A25E 2A 04 A0 423 CPIR HL
A261 ED B1 424 DEC HL
A263 2B 425 LD (#TEXTEND), HL
A264 22 06 A0 426 RET
A267 C9 427 #LOADING ", 00
A268 428 MMSTR3 DB "SAVING ", 00
A268 4C 4F 41 44 429 MMSTR4 DB "MEMORY OVER", 00
A26C 49 4E 47 20 430
A270 00 431
A271 53 41 56 49 432
A275 4E 47 20 00 433
A279 4D 45 4D 4F 434
A27D 52 59 20 4F 435
A281 56 45 52 00 436
A285 437

```


投稿プログラム大募集 のお知らせ

●あなたはどのようにしてプログラムに出会いましたか？

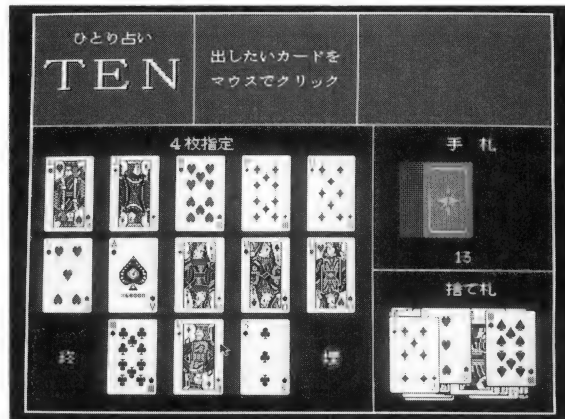
記号の羅列にすぎなかったプログラムリストが突然意味を持ったメッセージとして読み取れる、それを機に「プログラム」というものについてなにか納得できるようになる……。きっかけは雑誌のページの隅に載った小さな小さなプログラムだったのかもしれませんが。またはいくら見直してもエラーの出てくる長いBASICプログラムかもしれません。きっとそのプログラムにある「なにか」に魅かれてリストを打ち込んだことがあると思います。

あるソフトを使っている、なにかの記事を読んでいて、または突然に、「こんなソフトがあったらいいな」と思う。こういった小さな動機からプログラムは生まれてきます。あなたのアイデアを埋もれさせないでください。私たちはそこにある「なにか」を求めています。完成度の高いありふれたプログラムよりも、粗削りでもオリジナリティの光るプログラムのほうが、さらに誰かの「プログラム」を生むことになるはずです。

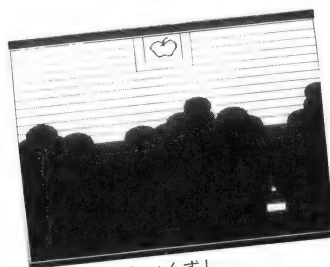
Oh!Xには毎月さまざまな投稿プログラムが掲載されています。これらのプログラムは、すべて読者の皆さんが日頃のパーソナルコンピューティングのなかで作り上げてきたものです。あなたも投稿プログラムを通じてOh!Xの誌面作りに参加しませんか？

●大作歓迎！

Oh!Xでは過去に40Kバイト程度のプログラムまで誌上に掲載した実績があります。また、どうしても誌面に載り切らない作品は付録ディスクに収録して配布したこともありました。どうぞ誌面には掲載できないからと諦めている方、とりあえずご連絡ください。



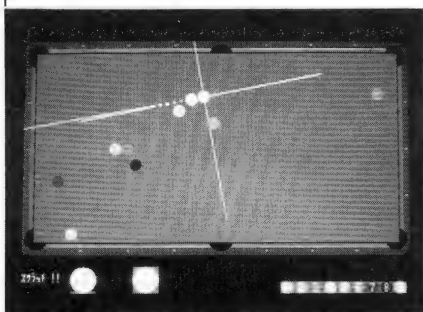
X68000用 カードゲーム TEN



X68000用 かべくずし

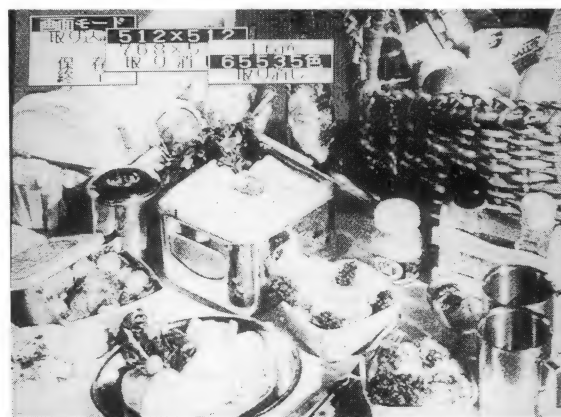
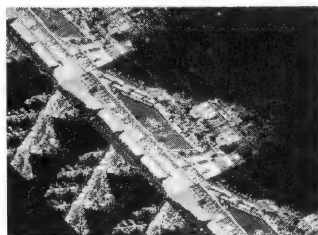


MZ-700用 Eyelarth



全機種共通システム用 BILLIARDS

X68000用 XROT0. X



X68000用 ハンディイメージスキャナアダプタの製作

投稿募集要項

- 1) お送りいただくプログラムには、住所、氏名、年齢、職業、連絡先電話番号、機種名、使用言語、動作に必要な周辺機器、マイコン歴などを明記のうえ、封書の宛先の最後には「Oh!X LIVE!」、「全機種共通システム」、「投稿ゲームプログラム」など、プログラムの内容を明確にご記入ください。
- 2) 投稿されるプログラムには詳しい内容を記入した原稿と一緒に変数表、メモリマップ、参考文献などもお書き添えのうえお送りください。また、お送りいただいた原稿については、当方で加筆修正をさせていただくことがあります。
- 3) お送りいただくプログラムは最低2回はセーブしておいてください。基本的に同封されたフロッピーディスク、カセットテープ、クイックディスクなどについてはご返送いたしませんので、あらかじめご了承ください。
- 4) ハード製作関係の投稿につきましては、最初は内容のわかる原稿のみお送りいただければ結構です。その後、当方で製作物が必要だと判断した場合には改めてご連絡いたします。

- 5) お送りいただいた作品の採用につきましては、掲載号が決定した時点で当方より連絡いたします。特にツール関係、ハード関係などのものにつきましては特集内容などを考慮したうえで採用決定されますので、結果を連絡するまでにかなり時間がかかる場合があります。
- 6) 投稿いただいたプログラムにバグなどが発見された場合は新しいプログラムの入ったメディアと一緒に文書にてご連絡ください。
- 7) 掲載されたプログラムに対しては当社規定の原稿料をお支払いいたします。また、投稿されたプログラムの著作権などは制作者に保留されますが、PDSなどとしてネットにアップロードされる場合は必ず事前に編集部までご連絡ください。なお、一般的モラルとして、他誌との二重投稿または他誌に掲載されたプログラムの移植などについては固くお断りいたします。

宛先

〒108 東京都港区高輪2-19-13 NS高輪ビル

ソフトバンク株式会社 Oh!X編集室「投稿プログラム」係



メニューによるファイル管理

亀田 雅彦 Kameda Masahiko

わかりやすいメニュー形式でファイル操作や実行を行うユーティリティです。いろいろと工夫されているので機能も豊富。これまでに作ったKAME-DOS用のコマンドなども起動できます。

今月はMENUだ

MENUといっても、Human Ver.2.0 についてるようなのじゃなくて、「ファイルセクタ」みたいなやつです。「COMMAND.X1」ではファイル名をキーボードから入力しますが、それを「カーソルで選択できるようにした」という代物です。

X68000のPDSでよく見かけていて、ちょっと作りたくなってしまう。簡単にできるかなと思ってたけど、意外にこずったかな。自分で作ったKAME-DOSながらも中身を忘れてるし、しかも、BASICのメモリ不足には泣かされるし。でも、それだけにかなりの力作だと思えます。操作性はほぼ完璧でしょう。はっはっは（自我自賛モード）。

余談はこのくらいにして、今回はX1シリーズ全機種対応です。機能は同じですが、プログラムはノーマルX1用とturbo用に分かれているので、打ち込むときは注意してください。もちろん実行には、いままでの外部コマンドと同様、「INTEGRAL X一式」が必要です。

機能

まず、機能から説明しましょう。「MENU.X1」は、「COMMAND.X1」の代表的な機能の一部をより簡単に実行するために作られました。したがって「COMMAND.X1」に完全に置き換わるものではなく、補佐的役割をはたします。それに、ファイルが4つに分かれるため「機動性が悪い」という面もあります。詳しくはまたあとで。

●ファイルセレクト実行

「COMMAND.X1」には、外部コマンドおよび普通のBASICファイルを、コマンドラインから実行する機能がありました。そのファイル名入力部分を、カーソル選択

で行います。また、パラメータとして渡すファイル名もカーソルでセレクトするので、キー入力する必要はありません。

●ドライブ/ディレクトリの移動

ドライブはポップアップ式メニューで、ディレクトリはファイル名と同じようにセレクトして移動します。画面には常にカレントディレクトリのファイル名一覧が表示されていて、スクロールさせて見ることができます。

●COPY, DEL

それぞれ対象ファイルをセレクトしておいてから、命令を実行します。選択方法はワイルドカード的な選択だけでなく、複数の任意のファイルをチョイスできるようにしたので、便利に使えると思います。

* * *

その他、ディスクタイプ自動判別、コンバート機能などは、「COMMAND.X1」と同様にサポートしました。逆にサポートされていないのは、デバイスドライバ、リダイレクション、TYPE命令などです。

結局、価値はどのへんにあるのか？ というと、「片手でジュースを飲みながら、マウスならぬテンキーで操作ができること」でしょう。あと、ファイルの実行が簡単なので、いろんな外部コマンドが考えられるようになったことかな。

入力方法

リスト1：共通・BASIC

リスト2：共通・BASIC

リスト3：turbo用・マシン語

リスト4：turbo用・マシン語

リスト5：ノーマルX1用・マシン語

リスト6：ノーマルX1用・マシン語

turbo用というのはturboBASIC、Z-BASICで使用する場合。CZ8FB01 ver.1.0で使うときはturboでもX1用のリストを使います（以下同様）。

まず、リスト1、2はturbo、X1共通な

ので、それぞれのBASICで打ち込んでください。ただし、リスト2はX1turboの場合そのまま入力しますが、X1の場合は一部変更します。いままでの外部コマンドと同じように、リスト2の最後のDATA文が英語と日本語で対になっています。X1ならこの英語のDATA文の注釈（'）をはずして、日本語のDATA文の部分は入力しないようにします。その入力結果を図1に示しておきます。それからX1の場合、すべての注釈行のコメントは入力しないでください（注釈行自体は入力します）。

上記のようにリスト1、2を打ち込んだら、

リスト1：SAVE "ME.BAT"

リスト2：SAVE "MENU.X1"

というファイル名でセーブします。

マシン語リストはX1turbo用、X1用それぞれ2つずつあります。X1turboならリスト3、4、X1ならリスト5、6をそれぞれのマシン語入力ツールから入力します。チェックサムを確認したら、

CLEAR &HC200 (turbo)

または、

CLEAR &HF200 (X1)

を実行して、

リスト3：SAVEM "MENU.OBJ",
&HC200,&HCAD1

リスト4：SAVEM "KEY.OBJ",&
HCE80,&HCEFF

または、

リスト5：SAVEM "MENU.OBJ",
&HF200,&HFAD6

リスト6：SAVEM "KEY.OBJ",&
HFE80,&HFEFF

としてセーブしてください。

最終的にBASIC2つ、マシン語2つの計4つのファイルができます。この4つのファイルは一緒に、同一のルートディレクトリにセーブするようにしてください。

なお、テストRUNするときには、
・必ずKAME-DOSの「COMMAND.X

1」を起動してから行う

・変数名ひとつ違うだけでも暴走したりファイルを壊したりする可能性がある
などの注意点は、前回までの外部コマンドと同じです。バックナンバーもよく読んでおいてください。

使い方

●起動方法

まずはKAME-DOSを立ち上げます。

「ME. BAT」
「MENU. X1」
「MENU. OBJ」
「KEY. OBJ」

の4つのファイルが揃っていることを確認したら、コマンドラインから、

ME [RET]

と打ち込んでください。順にファイルがロードされて、図2に説明されているような画面になります。「ME A:」「ME ファイル名」のように、パラメータやオプションをつけることはできません。また、「MENU. X1」には「X1」の拡張子がついていますが、最初の起動/再起動にかかわらず「ME. BAT」から立ち上げてください。

●ファイルセレクト

起動直後の状態では、反転カーソルがファイル名上にあります。もしこのときエラーがあればエラーメッセージを表示します。このメッセージはなにかキーを押せば消えますが、ドライブ変更などをしない限りエラー状態は変わりません。

テンキーの「8」「2」で、そのカーソルを上下に動かします。ファイルが多くてWINDOWに入りきらないときは、8・2キ

ーでスクロールします。また、「7」「1」キーを押すと、カーソルを上下に10ファイルずつ飛ばして移動できます。このファイルセレクトが、このプログラムでの最重要作業になります。

●ドライブセレクト

テンキーの「4」「6」を押してみてください。ファイル名WINDOWの上に、ポップアップ式にドライブセレクトメニューが表示されました。そのまま4、6を使って反転カーソルを左右に動かし、目的のドライブの上で4・6以外のキーを押してください。そのドライブのファイル名一覧が表示されます。

また、そのまま4、6を押し続けてメニューの外側に出せば、ドライブセレクトメニューはキャンセルされます。そのほかに、直接「A」「F」「W」「Y」を押してドライブを変更することもできます。

なおこのプログラムでは、A～F、W～Yの計9つのドライブしかサポートしていません。それぞれのドライブがどの物理デバイスなのかは、バックナンバーを見てください。

●ディレクトリ移動

ディレクトリ名は、ファイル名に混じって一緒に表示されます。そのディレクトリ名にカーソルを合わせて「リターンキー」を押せば下位ディレクトリに移動します。また、下位ディレクトリから上のディレクトリに戻るには、ファイル名の先頭にある「..」というファイル上でリターンキーを押します。テンキーの「.」を押すことで、直接戻する方法もあります。

●マークファイル

マークファイルとはファイル名を一時的に記憶しておいて、あとでCOPY,DEL,

実行などの対象ファイルとして使うものです。「COMMAND.X1」ではユーザーが1つひとつキータイプしていたファイル名を、まとめて指定できる利点があります。テンキーの「5」を押すと、ファイル名がマゼンタになり、頭に「<MRK>」がつきます。これがそのファイルをマークした印で、同時に複数個指定することができます。

ドライブ、ディレクトリを変更しても保存されますが、変更先で新たにマークすると元のマークは取り消されます。つまり、同時に同一のディレクトリにしか指定できないということです。左下の画面最下行に、マークファイルがあるかどうかを示します。「M」と出ているときは「マークがどこに残っている」、空白のときは「マークがない」ということです。

「+」キーで現在表示中の全ファイルをマーク。「-」キーはその逆、全マークを取り消します。

●集約機能

「/」（スラッシュ）キーには、3つの機能（COPY, DEL, ワイルドカード）が割り当ててあります。1回押せばCOPY, もう1回押すとDEL, 3回でワイルドカードの実行選択メニューがポップアップします。それぞれのときにリターンキーを押せば実行、スラッシュ・リターン以外のキーでキャンセルです。

1) COPY

COPYは以下の手順で行います。

- ・コピーしたいファイルをマークする（同一ディレクトリなら複数個可）
- ・コピー先となるドライブ&ディレクトリに移動する
- ・スラッシュを1回押して、COPYを実行する

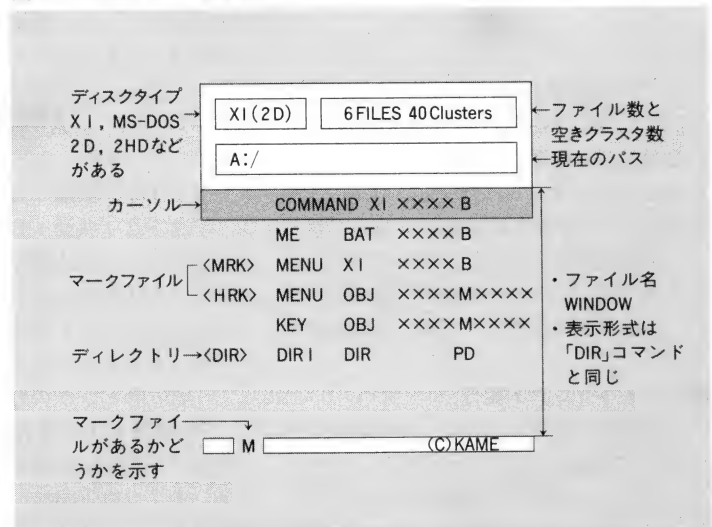
図1 リスト2のX1用変更点最終入力結果

```
3310 '
3320 '
3330 DATA COPY <mark> to current-dir
3350 DATA DELETE <mark> files
3370 DATA INPUT FILE-NAME for print
3390 DATA Error !!
3410 DATA Drive not ready
3430 DATA FILE Not Found
3450 DATA No exec file
3470 DATA Not REBOOT
```

図3 キー操作一覧表

8, 2.....ファイルセレクトのカーソル上下移動
7, 1.....8, 2の高速上下移動
4, 6.....ドライブセレクトのカーソル左右移動
5.....カーソル上ファイルをマーク
+,-.....全ファイルマーク, 全マーク解除
A~F.....ドライブ直接変換指定キー
W~Y.....
" ..".....上位ディレクトリへ戻る
" /".....COPY, DEL, ワイルドカード指定
RET.....ファイル名の上で押せばそのファイルを実行
スペース...MENU. X1の終了
*WIDTH40と80の切り替え

図2 MENU. X1 画面構成



「マークしたあと移動して COPY する」手順を間違えないでください。

2) DEL

DELの実行も同様です。

- ・消去したいファイルをマークする
- ・スラッシュを2回押し、DELを実行する

3) ワイルドカード

ワイルドカードを選択すると、ファイル名1行入力モードになります。ここでファイル名(ワイルドカードを含むことができる)を入力してください。現ディレクトリ内でそれに該当するファイル一覧を表示します。これと「+」キーのマーク機能を併用して、COPYなりDELなりすれば、ワイルドカード機能を実現できます。

●ファイル実行

COMMAND.X1では、ファイル名をキータイプしてファイルを実行していました。MENU.X1からは、実行したいファイルにカーソルを合わせて「リターンキー」を押すことで、そのファイルを実行します。また、パラメータとして渡すファイル名も、マークを利用して指定できます。方法は、

- ・「実行したいファイル」をマークする
- ・パラメータとして渡すファイルのあるディレクトリへ移動する
- ・そのファイル上でリターンキーを押す

例を挙げますと、「FORMAT.X1」「DISKCOPY.X1」などのようにファイル名を渡さない外部コマンドの場合、単純にそのファイル名の上でリターンキーを押します。

「GLOAD.X1」「XLOAD.X1」のよう

にファイル名を渡す場合(画像ファイルを「GAZO.GH1」とします)。まず「GLOAD.X1」をマークします。次に「GAZO.GH1」上でリターンキーを押します。これでコマンドラインから、

GLOAD GAZO.GH1

を実行したのと同じことになります。そして一度この作業を行えば、2度目からは「GAZO.GH1」上でリターンキーを押すだけで済みます(マークする必要がない)。前に実行したときのマークファイルを記憶しているからです。この記憶はなんらかの外部コマンドが実行されるまで保持されます。また次に「XLOAD.X1」などの、他のファイル名を渡す外部コマンドを実行するときは、マークするところからやり直してください。

なお、「GSAVE.X1」「XSAVE.X1」のように新規のファイル名を渡す外部コマンドに対して、それを指定するような機能はありません。つまり、これらの外部コマンドはMENU.X1からは実行できません。

●その他

「スペースキー」でMENU.X1を終了して、親プロセスへ戻ります。MENU.X1からMENU.X1を呼び出すような2重呼び出しはしないようにしてください。

「*」キーで「WIDTH 40 OR 80」の切り替えができます。例のごとく、どちらでも動くように設計されているので、見やすいほうで使ってください。「ME.BAT」には、起動時のWIDTHが設定されているの

で変更可です。画面の低/高解像度は、もとのKAME-DOSで設定されているものに従います。

最初に書いたとおり、このプログラムは4つのファイルで構成されているので、「COMMAND.X1」さえあれば動くという状態より「重く」なっています。それにコマンドラインのようにきめ細かな対応もできません。たとえばコピーの際、オプションがつけられない、コピー先のファイル名が指定できないなどです。つまり、MENU.X1とCOMMAND.X1は、利用目的に応じてユーザーが使い分けることを前提としているのです。

*

いろいろ書きましたが、感覚的にわかりやすい画面構成なので、とりあえず触ってみればすぐのみ込めます。大切なのは、前記の「マークファイル」「集約機能」「ファイル実行」ぐらいで、ほかは一読すれば十分でしょう。キー操作の一覧表も載せておくので、忘れてしまったときにでも使ってください。

今月は実用本位の設計思想なんです、操作性や機能など満足していただけたでしょうか? BASICの変数名にまで気を使って高速化しているので、小気味よいテンポは間違いないと思います。それから、今回のプログラム作成に当たって、X68000のいくつかのPDSを参考にさせていただきました。この場を借りてお礼申し上げます。

リスト1

```
1000 'MENU.X1 Ver 1.0 By Kameda
1010 '
1020 IF PEEK(&HD07F) THEN s=&HC200: CLEAR s: KLIST 0 ELSE s=&HF200
1030 '
1040 DEFUSR1=m_opens: DEFUSR2=m_preop: DEFUSR3=m_tranr
1050 '
1060 LOADM "MENU.OBJ": LOADM "KEY.OBJ"
1070 DEFUSR0=s+6: DEFUSR4=s+&HC: DEFUSR5=s+&H18
1080 mc=s : cp=s+3 : fw=s+9 : un=s+&HF
1090 mk=s+&H12: ck=s+&H15: tw=s+&H1B: cl=s+&H1E
1100 am=s+&H21: ac=s+&H24: kn=s+&H27: dl=s+&H2A
1110 co=s+&H2D
1120 '
1130 i0=s+&H40: du=s+&H68: cr=s+&H69: cb=s+&H6A
1140 de=s+&H6B: ys=s+&H6C: uu=s+&H6D: xx=s+&H6E
1150 wi=s+&H6F: ww=s+&H70: us=s+&H75: yy=s+&H76
1160 nw=s+&H78: ss=s+&H7A: c2=s+&H7C
1170 '
1180 i1=s+&HC58: i2=s+&HC80: i3=s+&HB48: sfe$="": CALL cl
1190 '
1200 IF width<80 THEN bxx=0 ELSE bxx=basx
1210 byy=0: POKE xx, bxx: POKE yy, byy
1220 POKE us, 5: POKE ys, 17: POKE uu, 1: POKE wi, width
1230 CONSOLE 0, 24: GOSUB 3030
1240 POKE v_dn, PEEK(s_dn): POKE v_ddrv+1, 7, 1: KEY0, ""
1250 '
1260 'MAIN
1270 fe$=""
1280 od=1: sb=0: op=1: GOSUB 2970
1290 es=0: k=PEEK(v_stop): t=PEEK(v_csd+PEEK(v_dn))
1300 st=t: sd=PEEK(v_dn): sk=sd: IF sk>21 THEN sk=sk-15
1310 a40=i1+sk*4+t: MEM$(s, 4)=MKI$(a40)
1320 IF k=3 AND t=0 THEN es=1
1330 IF k>0 AND k<>3 THEN es=2
1340 POKE uu, 1: POKE du, 1
1350 IF t>0 THEN POKE uu, 0: POKE du, 0 ELSE POKE a40+1, 0, 0, 0
1360 POKE cr, PEEK(du): POKE cb, 1: POKE c2, PEEK(du): GOSUB 2770
1370 IF es ELSE CALL cp
1380 REPEAT: d$=INKEY$: UNTIL d$<>"": KEY0, "": k=ASC(d$)
```

```

1390 ON PEEK(i2+k)-90 GOTO 1500,1530,1530,2330,2080,1670
1400 IF es GOTO 1380
1410 ON PEEK(i2+k) GOTO 1460,1470,1480,1490,1500,1570,1550,1620,1640,1650
1420 GOTO 1370
1430 GOTO 1280
1440 GOTO 1260
1450 'f1
1460 d$=USR0(CHR$(1)):GOTO 1420
1470 d$=USR0(CHR$(&HFF)):GOTO 1420
1480 d$=USR0(CHR$(10)):GOTO 1420
1490 d$=USR0(CHR$(&HF6)):GOTO 1420
1500 '
1510 IF k=>ASC("a") THEN k=k-ASC("a") ELSE k=k-ASC("A")
1520 POKE v_dn,k:POKE s_dn,k:GOTO 1440
1530 '
1540 KEY0,d$:GOSUB 2140:IF s OR es THEN 1440 ELSE 1420
1550 '
1560 IF st GOSUB 1960:GOTO 1440 ELSE 1420
1570 '
1580 CALL fw:IF PEEK(ww+3)=4 GOSUB 1910:GOTO 1440
1590 CALL kn:IF PEEK(v_yen)<>32 GOTO 1730
1600 IF PEEK(ww+3)=1 GOTO 1800
1610 IF fe$(0)=" " GOSUB 3230:GOTO 1420 ELSE 1700
1620 '
1630 CALL mk:d$=USR0(CHR$(1)):CALL kn:GOTO 1420
1640 CALL am:GOTO 1660
1650 CALL ac
1660 CALL kni:s=i3+116:fe$=MEM$(s+1,PEEK(s)):GOTO 1430
1670 '
1680 wids=- (wids=>80)*40-(wids<80)*80:CLS:WIDTH wids:GOTO 1190
1690 '
1700 'ex
1710 POKE ww+2,1:d$=fe$(0)
1720 s=i3+16:fe$(1)=MEM$(s+1,PEEK(s))+MEM$(i0+7,17):GOTO 1850
1730 '
1740 s=i3+16:fe$(1)=MEM$(s+1,PEEK(s))+MEM$(i0+7,17)
1750 POKE cr,1:CALL tw:IF PEEK(v_stop) GOTO 1440
1760 IF PEEK(ww+3)<>1 GOSUB 3230:GOTO 1440
1770 k=PEEK(v_mac):t=PEEK(v_dn):GOSUB 2010
1780 IF PEEK(&HD07F) THEN s=13+144:d$=d$+MEM$(s+3,PEEK(s)-2)
1790 fe$(0)=d$+MEM$(i0+7,17):GOTO 1840
1800 '
1810 fe$=MEM$(i0+7,17):GOSUB 2990:IF PEEK(v_stop) GOTO 1440
1820 k=PEEK(v_mac):t=sd:GOSUB 2010:fe$(0)=" "
1830 IF PEEK(&HD07F) THEN d$=d$+y$+dir$(t)
1840 d$=d$+MEM$(i0+7,17)
1850 proces$(proces)="MENU.X1":proces=proces+1
1860 CONSOLE 0,24:LOCATE 0,23
1870 IF PEEK(&HD07F) THEN CLEAR &HD000
1880 IF PEEK(ww+2)=0 GOTO 1900
1890 CHAIN d$
1900 KEY0,CHR$(26)+"RUN"+CHR$(34)+d$+CHR$(13):END
1910 '
1920 fe$=MEM$(i0+7,13):IF PEEK(cr)=0 GOTO 1960
1930 POKE v_cdf,1:GOSUB 2990:POKE v_cdf,0
1940 d$=MEM$(v_p256+1,PEEK(v_p256))
1950 k=PEEK(v_dn):dir$(k)=dir$(k)+d$:RETURN
1960 '
1970 fe$="":k=PEEK(v_dn):s=LEN(dir$(k))
1980 w=1:REPEAT:t=w:w=INSTR(t+1,dir$(k),y$):UNTIL w=s
1990 IF t=1 THEN dir$(k)=" " ELSE dir$(k)=LEFT$(dir$(k),t)
2000 POKE a40,0:POKE v_csdire+k,st-1:RETURN
2010 '
2020 d$=STR$(t)+": "
2030 IF t<4 THEN IF k=1 THEN DEVICE d$+"2" ELSE DEVICE d$+"0"
2040 IF t=4 THEN d$="MEM0:"
2050 IF t=5 THEN d$="MEM1:"
2060 IF t>21 THEN d$="EMM"+RIGHT$(STR$(t-22),1)+": "
2070 DEVICE d$:RETURN
2080 '
2090 d$=USR3(proces$(proces-1)):IF PEEK(v_stop) GOSUB 3240:GOTO 1440
2100 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+": "+RIGHT$(STR$(3-PEEK(v_mac))),1)
2110 CONSOLE 0,24:LOCATE 0,23:proces=proces-1
2120 IF PEEK(&HD07F) THEN CLEAR &HD000 ELSE CLEAR
2130 CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
2140 '
2150 d$=USR4(CHR$(1,7,38,4)):i=bx+2:w=by+9
2160 LOCATE i+12,w-1:COLOR 5:CREV 1:PRINT "*DRIVE SELECT*";
2170 COLOR 7:LOCATE i,w
2180 PRINT " A: B: C: D: E: F: W: X: Y:":CREV 0
2190 s=PEEK(v_dn)+1:IF s>22 THEN s=s-16
2200 t=i+4*(s-1)
2210 LOCATE t,w:PRINT SCRN$(t,w,4);
2220 REPEAT:d$=INKEY$:UNTIL d$<>"":KEY 0,""
2230 IF d$<>"4" AND d$<>"6" GOTO 2290
2240 LOCATE t,w:CREV 1:PRINT SCRN$(t,w,4):CREV 0
2250 IF d$="4" THEN t=t-4:s=s-1
2260 IF d$="6" THEN t=t+4:s=s+1
2270 IF s<1 OR s>9 GOTO 2310
2280 GOTO 2210
2290 IF s>6 THEN s=s+16
2300 CALL un:POKE v_dn,s-1:POKE s_dn,s-1:RETURN
2310 CALL un:s=0:RETURN
2320 '
2330 'ESC
2340 w=1:RESTORE 3320
2350 d$=USR4(CHR$(4,7,32,5)):s=bx+4:t=by+7
2360 READ d$:LOCATE s+2,t+1:CREV 1:COLOR 5
2370 PRINT LEFT$(d$+STRING$(28," "),28):COLOR 7:CREV 0
2380 LOCATE s+11,t+3:CREV 1:COLOR 6:PRINT "Yes:[RET]":COLOR 7:CREV 0
2390 KEY0,"":REPEAT:d$=INKEY$:UNTIL d$<>" "
2400 IF d$=CHR$(47) THEN w=w+1:IF w<4 GOTO 2360
2410 IF d$<>CHR$(13) THEN w=0:CALL un:GOTO 1420
2420 LOCATE s+2,t+3:PRINT STRING$(28," "):LOCATE s+2,t+3

```



```

2430 ON w GOTO 2480,2460,2450
2440 '
2450 k=28:GOSUB 2710:GOTO 1430
2460 GOSUB 2510:IF PEEK(v_stop) GOSUB 3200
2470 GOTO 1440
2480 dn=PEEK(v_dn):GOSUB 2600:POKE v_dn,dn
2490 POKE v_ddrv+1,7,1:IF PEEK(v_stop) GOSUB 3200
2500 GOTO 1440
2510 'del
2520 x=POS(0):y=CSRLIN:POKE cr,1
2530 GOSUB 2680:IF PEEK(v_stop) OR PEEK(cr)>127 RETURN
2540 POKE cr,1
2550 LOCATE x+1,y:PRINT MEM$(i0+7,17);
2560 CALL d1:IF PEEK(v_stop) RETURN
2570 IF PEEK(cr)<=127 GOTO 2550
2580 POKE v_stop,0:CALL m_clos2
2590 RETURN
2600 'copy
2610 POKE cr,1:t=0:s=&H10:w=POS(0):i=CSRLIN
2620 '
2630 POKE v_ddrv+1,7,1
2640 GOSUB 2680:IF PEEK(v_stop) OR PEEK(cr)>127 RETURN
2650 LOCATE w+1,i:PRINT MEM$(i0+7,17);
2660 CALL co:IF PEEK(v_stop) RETURN
2670 GOTO 2620
2680 'SUB
2690 CALL tw:k=PEEK(v_dn):IF PEEK(v_csdire+k)=0 THEN dir$(k)="
2700 RETURN
2710 '
2720 x=POS(0):y=CSRLIN:CONSOLE y,1,x,k
2730 KEY0,sfe$:fse$="":INPUT "",fse$
2740 CONSOLE 0,24
2750 sfe$=fse$:RETURN
2760 '
2770 'INIT
2780 CALL m_dirsb:POKE de,PEEK(v_yen)
2790 MEM$(nw,2)=MEM$(v_bf,2)
2800 d$=CHR$(65+sd)+":"+y$+dir$(sd)
2810 k=LEN(d$):MEM$(i3+16,k+1)=CHR$(k)+d$
2820 k=LEN(fse$):MEM$(i3+116,k+1)=CHR$(k)+fse$
2830 MEM$(i3+141,3)=CHR$(PEEK(v_yen))+MEM$(v_fors,2)
2840 w$=dms$(PEEK(v_mac)):s=INT((11-LEN(w$))/2)
2850 LOCATE bxx+1,byy+1:PRINT STRING$(11," ");
2860 LOCATE bxx+s+1,byy+1:PRINT w$;
2870 LOCATE bxx+13,byy+1:PRINT STRING$(26," ");
2880 LOCATE bxx+13,byy+1:PRINTUSING "####",PEEK(v_yen);
2890 COLOR 4:PRINT " FILES ":COLOR 7
2900 PRINTUSING "####",CVI(MEM$(v_fors,2));
2910 COLOR 4:PRINT " Clusters":COLOR 7
2920 LOCATE bxx+2,byy+3:COLOR 6:PRINT LEFT$(d$+STRING$(37," "),37):COLOR 7
2930 LINE (bxx,byy+PEEK(us))-(bxx+40,byy+PEEK(us)+PEEK(ys)),",",bf
2940 CALL kn:ON es GOTO 3220,3210
2950 d$=USR0(CHR$(PEEK(a40))):CALL mc:RETURN
2960 '
2970 'open
2980 d$=USR5(CHR$(1,od,sb,op)+fse$):GOTO 3000
2990 d$=USR5(CHR$(2,1)+fse$)
3000 k=PEEK(v_dn):IF PEEK(v_csdire+k)=0 THEN dir$(k)="
3010 RETURN
3020 '
3030 '
3040 CLS:CGEN 1:d$=STRING$(11,"1"):w$=STRING$(26,"1")
3050 s=byy+PEEK(us)+PEEK(ys)+1
3060 LOCATE bxx,s:PRINT "0112 0";STRING$(27,"1");CHR$(33,34,35,36,37);"1";
3070 LOCATE bxx,byy :PRINT "5";d$;"8";w$;
3080 LOCATE bxx,byy+1:PRINT "4";STRING$(11," ");"4";STRING$(26," ");
3090 LOCATE bxx,byy+2:PRINT ":";d$;"9";w$;
3100 LOCATE bxx,byy+3:PRINT "4";STRING$(38," ");
3110 LOCATE bxx,byy+4:PRINT "3";STRING$(38,"1");
3120 LINE (bxx+39,s)-(bxx+39,s),"2"
3130 LINE (bxx+39,byy )-(bxx+39,byy ),"6"
3140 LINE (bxx+39,byy+1)-(bxx+39,byy+3),"4"
3150 LINE (bxx+39,byy+2)-(bxx+39,byy+2),","
3160 LINE (bxx+39,byy+4)-(bxx+39,byy+4),"7"
3170 CGEN 0:RETURN
3180 '
3190 '
3200 RESTORE 3380:w=15:GOTO 3250
3210 RESTORE 3400:w=8:GOTO 3250
3220 RESTORE 3420:w=8:GOTO 3250
3230 RESTORE 3440:w=8:GOTO 3250
3240 RESTORE 3460:w=8
3250 READ m$:s=LEN(m$):t=INT((40-s)/2)
3260 d$=USR4(CHR$(t-2,w,s+4,3))
3270 LOCATE bxx+t,byy+w+1:CREV 1:PRINT m$;:CREV 0
3280 KEY0,"":REPEAT:d$=INKEY$:UNTIL d$<>"":KEY0,d$
3290 CALL un:POKE v_stop,0:RETURN
3300 '
3310 '
3320 'DATA COPY <mark> to current-dir 'X1用 'を取りturbo用の行は打ちこまない
3330 DATA マークファイルをコピーします 'turbo用 turboはそのまま打ちこむ
3340 'DATA DELETE <mark> files ** 以下同じ **
3350 DATA マークファイルを消去します
3360 'DATA INPUT FILE=NAME for print
3370 DATA ファイル名を入力してください
3380 'DATA Error !!
3390 DATA エラーが発生しました!!
3400 'DATA Drive not ready
3410 DATA ドライブの準備ができていません
3420 'DATA FILE Not Found
3430 DATA ファイルがありません
3440 'DATA No exec file
3450 DATA 実行不可能なファイルです
3460 'DATA Not REBOOT
3470 DATA リブートできません

```

```

1000 'ME.BAT Ver 1.0 By Kameda
1010
1020 widts=80 '80 or 40
1030 basx=20 '0<=basx<=40 (only widts=80)
1040 ,
1050 WIDTH widts:SCREEN:f$$(0)="
1060 RESTORE 1080:REPEAT:READ i,d$:DEFCHR$(i)=HEXCHR$(d$):UNTIL i=59
1070 CHAIN "MENU.X1"
1080 ,
1090 DATA 32,000000000000000000000000000000000000000000000000000
1100 DATA 33,00BD665A5E5A6BDB0EBD000000000000000000BD665A5EA66000
1110 DATA 34,00FF9DDBD7CDBCDF00F00000000000000000FF9DDBD7CBDC0000
1120 DATA 35,00FFFF9F5EDDDBB8FF00FF0000000000000000FF9F5EDDDBB800
1130 DATA 36,00FFF7856ED7D7DF00F0000000000000000FFF7856ED7D7D000
1140 DATA 37,00FFF03BF83BF81FF00F0000000000000000FFF03BF83BF81000
1150 DATA 48,007F7F7F7F7F7F7F007F404040404040007F7F7F7F7F7F40
1160 DATA 49,00FFFFFFFFFFFFFFFF00FF0000000000000000FFFFFFFFFFFFFFFF
1170 DATA 50,00FFFFFFFFFFFFFFFF00FE0000000000000000FEFFFFFFFFFEFE00
1180 DATA 51,7F7F7F7F7F7F7F7F7F40414040404040407E7F7F7F7F7F7F40
1190 DATA 52,7F7F7F7F7F7F7F7F7F40404040404040407E7E7E7E7E7E7E
1200 DATA 53,007F7F7F7F7F7F7F007F404040404040007F7F7F7F7F7F7E
1210 DATA 54,00FFFFFFFFFFFFFFFF00FE0000000000000000FEFFFFFFFFFEFE00
1220 DATA 55,7FFFFFFFFFFFFFFFFF40C00000000000007EEFEFEFEFEFEFE00
1230 DATA 56,00FFFFFFFFFFFFFFFF00F00000000000000000FFFFFFFFFFFFFFFF
1240 DATA 57,7FFFFFFFFFFFFFFFFF40C00000000000007EFFFFFFFFFFFFFF00
1250 DATA 58,7F7F7F7F7F7F7F7F40414040404040407E7F7F7F7F7F7E
1260 DATA 59,7FFFFFFFFFFFFFFFFF40C00000000000007EFFFFFFFFFEFE3E

```


C200	C3	DA	C6	C3	00	C7	C3	3D	:	ED
C208	C7	C3	AD	C7	C3	97	F5	C3	:	E0
C210	58	C6	C3	6C	C5	C3	96	C5	:	A0
C218	C3	6D	CA	C3	DA	C3	C3	BE	:	D5
C220	C3	C3	35	C5	C3	95	C9	C3	:	64
C228	76	C3	C3	51	C3	C3	89	C2	:	1E
C230	94	20	50	41	54	48	20	65	:	66
C238	72	72	6F	72	00	11	00	8B	:	8E
C240	0B	94	20	42	61	64	20	52	:	38
C248	45	43	4F	52	44	00	16	00	:	83
C250	C2	08	94	20	52	65	73	65	:	10
C258	72	76	65	64	20	66	75	74	:	20
C260	75	72	65	00	0F	00	CC	0B	:	35
C268	00	00	00	00	14	01	00	50	:	65
C270	45	43	00	07	00	03	00	00	:	92
C278	00	00	00	00	00	00	00	27	:	27
SUM:	22	F5	7E	A1	76	C8	AD	D2	:	C540

C2B0	00	04	00	0B	94	20	00	DD
C2B8	00	AF	32	87	C2	3E	10	32
C2C0	88	C2	3A	97	E0	32	7D	C2
C2C8	3E	07	32	8B	D1	32	8C	D1
C2A0	2A	78	E0	32	7F	C2	2A	7A
C2A8	E0	22	81	C2	3E	03	32	83
C2B0	C2	3E	07	32	84	C2	3E	01
C2B8	32	85	C2	3E	03	32	86	C2
C2C0	11	83	C2	06	04	CD	18	C2
C2C8	3A	8C	E0	B7	C0	3A	97	E0
C2D0	32	7E	C2	CD	0B	C3	CD	22
C2D8	C3	3A	88	C2	32	05	EC	AF
C2E0	32	9C	E0	C0	0C	D0	3A	8C
C2E8	E0	B7	C0	3A	9C	E0	B7	20
C2F0	F2	CD	3A	87	C3	3A	B1	D0
C2F8	82	D0	3A	39	C2	B7	28	32

SUM: 8A C0 C2 84 67 86 AA D9 FC4A

C300	32	D2	30	E2	02	32	80	D1	47
C308	33	D5	00	3A	7D	C2	E6	01	98
C310	C0	3A	7E	C2	E6	01	C8	3E	29
C318	04	C0	32	87	C2	3E	12	32	88
C320	C2	C9	3A	7D	C2	E6	01	C8	33
C328	3A	7E	C2	E6	01	C0	3E	20	B7
C330	32	87	C2	3E	12	32	88	C2	48
C338	C9	3A	E5	EC	B7	C0	3A	87	00
C340	C2	FE	20	C8	2A	7F	C2	D2	35
C348	86	D1	2A	81	22	22	88	D1	3F
C350	C9	CD	03	0E	CD	12	D0	CD	F5
C358	15	C2	3A	69	C2	FE	80	D0	8A
C360	11	5A	E0	21	47	C2	01	0D	7D
C368	00	ED	B0	21	55	C2	01	03	D9
C370	00	ED	B0	C3	06	D0	3A	6E	DE
C378	C2	C6	04	5F	16	00	3A	7E	B1

SUM: A9 5D 13 7F 63 A4 71 4D 4942

C380	C2	71	C2	86	21	63	C2	51	FF
C388	86	67	26	00	CD	63	C4	85	FE
C390	55	06	08	21	48	CD	7E	23	AA
C398	B7	20	16	10	F9	16	20	18	34
C3A0	82	16	40	7A	32	84	D1	C1	27
C3A8	3E	20	08	47	3E	03	ED	79	00
C3B0	3E	18	80	47	AF	ED	79	78	AA
C3B8	D6	08	47	ED	51	C9	11	49	86
C3C0	D2	14	8D	C9	11	37	01	AF	EE
C3C8	77	DD	B0	C9	11	49	CD	21	25
C3D0	48	CD	01	0F	00	AF	77	ED	38
C3D8	B0	C9	DD	21	80	EE	3E	01	24
C3E0	DD	77	00	DD	77	01	DD	77	FE
C3E8	03	AF	DD	77	02	11	84	EE	8D
C3F0	21	D8	CD	63	C4	21	3C	17	E5
C3F8	CE	CD	63	C4	CD	58	C4	3A	1E

SUM: 23 48 69 B9 72 59 7E 5B 8BB5

C400	8C	00	B7	C2	C3	CD	2A	68
C408	D0	3A	81	D1	32	6B	22	E5
C410	91	E0	22	78	C2	CD	15	C2
C418	CD	48	C4	3A	69	C2	FE	80
C420	D0	DD	21	80	EE	3E	01	DD
C428	77	00	DD	77	01	DD	77	02
C430	AF	DD	77	03	11	84	EE	21
C438	D8	CD	CD	63	C4	21	47	C2
C440	01	11	00	ED	B0	C3	5B	C4
C448	3A	84	D1	21	55	CE	BE	C8
C450	3E	80	32	69	C2	C3	CC	43
C458	EB	11	80	EE	B7	ED	52	C5
C460	C3	18	C2	7E	B7	C4	13	06
C468	00	23	ED	B0	C9	1A	4F	95
C470	FE	02	28	25	FE	C3	28	2A
C478	CD	C2	C4	CD	D3	C4	C5	D5

SUM: 7A EE 81 27 BC 67 D2 F6 3A57

C480	CD	21	D0	D1	C1	3A	8C	0E	F6
C488	97	00	68	3A	84	D1	47	7D	32
C498	00	6F	26	00	19	EB	C3	24	10
C4B0	00	1A	85	32	80	D1	C3	48	00
C4A0	21	D0	CD	C2	C4	CD	E6	4A	BB
C4A8	CD	D3	C4	3A	58	CD	47	11	1B
C4B0	59	CD	CD	21	D0	CD	F6	4A	6B
C4B8	3A	8C	0E	B7	0C	06	00	C3	E6
C4C0	24	D0	C1	3A	8B	0E	4F	06	B3
C4C8	00	05	0C	EC	09	7E	32	97	1D
C4D0	E0	C1	C9	1A	32	80	D1	13	1A
C4D8	05	1A	32	65	0E	13	05	1A	1A
C4E0	32	81	D1	13	05	C9	C5	D5	FF
C4E8	21	68	E0	11	DD	EE	01	23	69
C4F0	00	ED	B0	D1	C1	C9	C5	D5	92
C4F8	11	68	E0	21	DD	EE	01	23	69

SUM: D2 70 70 9F 62 42 6D 5A F5CD

C500	00	D0	B0	D1	C1	C9	11	69	:	72
C508	C2	14	FE	01	20	09	32	6A	:	95
C510	C3	24	78	C2	02	00	CA	CD	:	98
C515	A3	C9	A6	20	B8	1A	21	55	:	C0
C520	CE	BE	30	0C	3C	12	18	E1	:	05
C528	2F	A6	77	CD	09	C2	B7	C9	:	64
C530	3E	80	12	37	C9	CD	7B	C9	:	4E
C535	30	0E	11	D8	CD	21	58	CD	:	3A
C540	01	80	00	ED	B0	CD	95	C9	:	4F
C548	3E	01	32	69	CD	2	3A	55	:	CE
C550	47	C5	CD	09	C2	3A	73	C2	:	13
C558	FE	04	28	08	11	69	C2	CD	:	31
C560	A3	C9	B6	77	21	69	C2	FE	:	1E
C568	1	C1	10	E6	C9	3A	73	C2	:	FE
C570	04	C8	CD	7B	C9	30	0E	11	:	20
C578	D8	CD	21	58	CD	01	80	00	:	60

SUM: 56 A4 47 16 1F 25 01 9E 75D1

C580	ED	B0	CD	95	C9	11	69	C2	: 04
C588	CD	A3	C9	57	A6	20	04	7A	: E0
C590	B6	77	C9	2F	A6	77	C9	D5	: E0
C598	DD	E1	3A	6E	C2	DD	86	00	: 81
C5A0	5F	16	30	3A	76	C2	DD	86	: 44
C5A8	01	6F	26	00	CD	63	CA	ED	: 71
C5B0	43	B7	CA	DD	7E	02	32	B9	: 00
C5B8	CA	DD	7E	03	32	BA	CA	ED	: CF
C5C0	5B	DF	EC	AF	21	00	38	CD	: F8
C5C8	DB	C5	3E	20	21	20	30	CD	: 30
C5D0	DB	C5	3E	07	21	0F	20	CD	: 02
C5D8	DB	C5	C9	53	C2	BD	CA	78	: 51
C5E0	84	47	32	BC	CA	CD	2E	C6	: 40

```
C5E8 CD ED C5 C1 C9 03 DD 6E : 57
C5F0 02 ED 78 CD 18 E0 13 3A : 79
C5F8 BD CA ED 79 03 2D 20 F1 : 2E
```

SUM: B6 DD 94 01 0D 2F EF 68 5FD1

C600	ED	4B	B7	CA	3A	BC	CA	47	: C0
C608	2D	7E	02	6F	26	00	09	44	: 3B
C610	4D	10	DD	6E	03	E5	3A	6F	C2
C618	6F	26	00	09	44	4D	E1	2D	: 3D
C620	C8	ED	78	CD	18	40	E3	3A	: 3F
C628	BD	CA	ED	79	18	E6	7D	32	: 9A
C630	BB	CA	DD	6E	03	2D	66	02	: 1B
C638	E5	CD	78	CD	18	40	E3	3A	: 3C
C640	BB	CA	ED	79	03	25	20	F1	: 24
C648	C1	E5	3A	6F	C2	6F	26	00	: A6
C650	09	44	4D	E1	2D	66	02	C9	: 6F
C658	ED	4B	B7	CA	ED	5B	DF	EC	: CC
C660	3E	38	CD	70	C6	30	C6	C9	: B4
C668	70	C6	3E	2D	CD	70	C6	C9	: 60
C670	E5	C0	47	32	BC	CA	CD	B7	: C8
C678	C6	CD	7E	C6	C1	C9	03	3A	: 9E

SUM: 36 C3 DC E1 C3 16 F5 4F 9C78

C680	B9	A	6	F	D	15	E0	13	ED	B4
C682	79	03	2D	20	F6	ED	43	ED	AB	7E
C690	C4	3A	BC	A	4	7	3A	B9	CA	28
C698	6F	26	00	09	44	4	3A	BA	23	8E
C6A0	C4	6F	E5	44	6D	2	C	6F	26	1F
C6A8	00	09	34	4	1E	2D	C8	6D	26	3F
C6B0	15	E0	13	ED	79	18	EB	3A	AE	00
C6B8	BA	C4	6F	3A	B9	CA	67	C5	CD	00
C6C0	CD	15	E0	13	ED	79	03	25	63	9E
C6D0	20	P6	C1	E5	3A	6B	C2	6F	96	00
C6D8	2E	00	09	44	4D	1E	2D	20	FE	00
C6DR	E2	C9	3A	68	C2	4F	3A	6C	0A	00
C6E0	C2	81	4F	3A	6B	C2	B9	30	E2	00
C6E8	01	4F	3A	68	C2	47	11	70	70	00
C6F0	C2	79	B8	ED	78	12	13	AF	17	00
C6F8	12	1B	CD	A6	7	04	18	AF	74	00

SUM: 90 87 F5 32 BA 5C FB 7A 0E35

C700	D5	3A	68	C2	5F	3A	7C	C2	: 10
C708	B8	38	14	57	3A	6C	C2	83	: 45
C710	B8	38	0C	7A	11	70	C2	12	: C1
C718	13	AF	12	1B	CD	A6	C7	3A	: 63
C720	69	C2	11	70	C2	12	13	3C	: D1
C728	08	12	1B	CD	A6	C7	3A	6D	: 10
C730	C2	57	3A	69	C2	92	ED	5B	: 58
C738	7A	C2	12	D1	C9	65	D5	1A	: 90
C740	FE	80	D2	78	C7	4F	3A	69	: 81
C748	C2	81	32	69	C2	4F	3A	6B	: 94
C750	C2	B9	30	C2	32	69	C2	3A	: 45
C758	69	C2	4F	3A	68	C2	47	3A	: 51
C760	6C	C2	80	B9	D2	A3	C7	3A	: D1
C768	6C	C2	47	3A	69	C2	90	32	: 90
C770	68	C2	CD	00	C3	A3	C7	3A	: E0
C778	3A	6D	C2	47	3A	69	C2	4F	: 66

SUM: 6F 75 EB 7D C4 46 0F 7B 112

C780	1A	81	FE	80	30	08	B8	38	: 4
C788	05	32	69	C2	18	04	78	32	: 2
C790	69	C2	3A	68	C2	4F	3A	69	: 8
C798	C2	B9	D2	A3	C7	32	68	C2	: 1
C7A0	CD	00	C2	D1	C1	C9	CD	E1	: 9
C7A8	C8	CD	C6	C7	C9	D5	11	70	: 4
C7B0	C2	3A	69	C2	12	CD	E1	C8	: A
C7B8	3A	69	C2	B7	28	06	11	54	: A
C7C0	C2	3E	2E	12	D1	C9	C5	D5	: 7
C7C8	E5	3A	68	C2	4F	1A	91	6F	: B

C7D0 3A 75 C2 85 6F 3A 76 C2 : D7
C7D8 85 6F 26 00 13 1A 32 71 : EA
C7E0 C2 3A 6E C2 5F 16 00 CD : 6E
C7E8 63 CA CD 84 CA DD 21 B2 : F8
C7F0 CA 3E 28 DD 77 00 CD 10 : 61
C7F8 C8 21 40 C2 DD 70 02 CD : 97
SUM: F8 5D 47 9C B4 98 90 D5 564E

C800 55 C8 DD 46 02 23 03 DD : 45
C808 35 00 20 F0 E1 D1 C1 C9 : 81
C810 AF DD 77 03 DD 77 04 3A : 98
C818 71 C2 F6 07 DD 77 01 E6 : 6B
C820 08 C0 3A 73 C2 FE 04 3E : 77
C828 06 CA 51 C8 3A 74 C2 B7 : 10
C830 3E 03 C2 51 C8 3A 72 C2 : 8A
C838 B7 28 05 16 04 DD 72 03 : 50
C840 3A 73 C2 B7 C8 1E 05 FE : 0F
C848 03 28 02 1E 04 DD 73 04 : A3
C850 C9 DD 77 01 C9 3A B1 CA : 9C
C858 B7 28 10 F3 3E 1D D3 00 : 10
C860 7E CD 99 30 30 29 3E 1E : C9
C868 D3 00 FB DD 7E 00 DD 5E : 64
C870 01 DD 56 03 FE 13 CD D2 : E6
C878 C8 FE 12 CC D2 C8 FE 11 : 4D
SUM: 84 64 03 87 B6 C1 54 AB 7C6D

C880 CC D2 C8 FE 08 CC D8 C8 : D8
C888 7B 5E 16 00 C3 B5 C8 5E : 85
C890 23 5E 2B CD 81 2F CD B6 : AC
C898 2F DD B6 01 CD B5 C8 DD : EA
C8A0 46 02 03 DD 70 02 23 DD : 9A
C8A8 35 00 CB F2 CD B5 C8 3E : 7A
C8B0 1E D3 00 FB C9 F5 F5 DD : 7C
C8B8 7E 02 C6 20 47 F1 ED 79 : 04
C8C0 DD 7E 02 C6 38 47 ED 51 : E0
C8C8 DD 7E 02 C6 30 47 ED 59 : E0
C8D0 F1 C9 7A B7 C8 5A AF C9 : 85
C8D8 DD 56 04 7A B7 C8 5A AF : 39
C8E0 C9 C5 D5 E5 ED C3 BE CA : 10
C8E8 21 40 C2 11 41 C2 01 27 : 5F
C8F0 00 3E 20 77 ED B0 5B : 8A
C8F8 BE CA 1A B7 20 06 CD C0 : 0C
SUM: E0 6A A6 97 88 7D 5E 50 016E

C900 C9 C3 3D C9 FE 01 20 09 : BA
C908 2A 78 C2 22 C0 CA C3 2F : 02
C910 C9 5F 2A C0 CA 3A C2 : 42
C918 FE 01 20 06 2A 78 C2 22 : AB
C920 C0 CA 93 28 0A 57 DC CF : 51
C928 C9 D4 E5 C9 22 C0 CA 11 : 08
C930 47 C2 CD FE C9 ED 5B BE : A3
C938 CA 1A 32 6A C2 CD 4F C9 : 27
C940 CD 09 CA ED 5B BE CA 1A : 8A
C948 32 7C C2 E1 D1 C1 C9 AF : 5B
C950 32 74 C2 3A 60 C2 FE 44 : 06
C958 21 C8 CA 28 15 CD 7B C9 : 01
C960 D8 ED 5B BE CA CD A3 C9 : E1
C968 A6 C8 21 CD CA 3E 01 32 : 97
C970 74 C2 11 41 C2 01 05 00 : 50
C978 ED B0 C9 21 58 CD 11 D8 : 95
SUM: 85 FD 2E 27 B8 35 25 2C 6F68

C980 CD 06 80 1A BE 20 06 13 : 64
C988 23 10 F8 B7 C9 78 FE 04 : 25
C990 CD 95 C9 37 C9 11 49 CD : 61
C998 21 48 CD 01 0F 00 AF 77 : 6C
C9A0 ED B0 C9 1A CB 3F CB 3F : 94
C9A8 CB 3F 4F 06 00 21 48 CD : 95
C9B0 09 1A E6 07 0E 80 B7 28 : 7D
C9B8 05 CB 09 3D 18 F8 79 C9 : 68
C9C0 11 47 C2 3E 2E 12 13 12 : BD
C9C8 11 60 C2 3E 44 12 C9 F5 : 85
C9D0 7A ED 44 57 01 17 00 09 : 23
C9D8 CD 12 E0 23 FE 02 20 F8 : 05
C9E0 15 20 F4 F1 C9 F5 01 19 : F2
C9E8 00 B7 ED 42 CD F5 C9 23 : 94
C9F0 15 20 F6 F1 C9 CD 12 E0 : A4
C9F8 FE 0D C8 2B 18 F7 CD 12 : EC
SUM: 44 71 5C B2 38 77 E4 8E 937A

CA00 E0 FE 0D C8 23 12 13 18 : 13
CA08 F5 AF 32 72 C2 32 73 C2 : 71
CA10 3A 60 C2 FE 44 20 06 C3 : 02
CA18 04 32 73 C2 C9 21 C2 CA : E1
CA20 CD 42 CA 21 C5 CA CD 42 : 98
CA28 CA 16 03 3A 60 C2 FE 4D : 8A
CA30 28 0B 16 01 FE 42 28 05 : B7
CA38 16 02 FE 41 C0 7A 32 73 : 36

CA40 C2 C9 C5 11 55 C2 06 03 : 81
CA48 1A 13 CD 5A CA BE 23 20 : 1F
CA50 07 10 F5 3E 01 32 72 C2 : B1
CA58 C1 C9 FE 61 D8 FE 7B DD : 0A
CA60 D6 20 C9 3A 6F C2 FE 28 : 50
CA68 CA 78 CA 29 29 29 29 E5 : 95
CA70 29 29 19 D1 19 44 4D C9 : AF
CA78 29 29 29 E5 29 29 19 D1 : 9C
SUM: 7E 43 AF BA A7 D5 16 45 D3C6

CAB0 19 44 4D C9 C5 3A 77 C2 : AB
CAB8 B7 3E 00 20 1F 3A 7F D0 : BD
CA90 B7 28 19 F3 3E 1D ED 79 : AC
CA98 3A 00 00 4F 3A 00 10 47 : 1A
CAA0 3E 1E D3 00 FB 78 B9 3E : 99
CAA8 01 20 01 AF 32 B1 CA C1 : 3F
CAB0 C9 00 44 4E 8B E0 02 06 : CE
CAB8 56 5F 53 54 4F 50 00 00 : FB
CAC0 00 00 58 31 20 42 41 54 : 80
CAC8 3C 44 49 52 3E 3C 4D 52 : 34
CAD0 4B 3E
SUM: A6 C9 72 FF C1 68 06 FD 2BF2

リスト4

CE80 00 00 00 00 00 00 00 00 : 00
CE88 00 00 00 00 00 00 06 00 : 06
CE90 00 00 00 00 00 00 00 00 : 00
CE98 00 00 00 00 00 00 04 03 : 07
CEA0 5F 00 00 00 00 00 00 00 : 5F
CEA8 00 00 60 09 00 0A 07 5E : D8
CEB0 00 03 01 00 5C 08 5D 0A : C9
CEB8 02 00 00 00 00 00 00 00 : 02
CEC0 00 5B 5B 5B 5B 5B 5B 00 : 22
CEC8 00 00 00 00 00 00 00 00 : 00
CED0 00 00 00 00 00 00 00 5B : 5B
CED8 5B 5B 00 00 00 00 00 00 : 86
CEE0 00 5B 5B 5B 5B 5B 5B 00 : 22
CEE8 00 00 00 00 00 00 00 00 : 00
CEF0 00 00 00 00 00 00 00 5B : 5B
CEF8 5B 5B 00 00 00 00 00 00 : 86
SUM: 17 6F 17 BF 12 CE 1E 1B 2888

リスト5

F200 C3 D4 F6 C3 FA F6 C3 37 : 3A
F208 F7 C3 A7 F7 C3 97 F5 C3 : 6A
F210 55 F6 C3 6C F5 C3 06 F5 : 2D
F218 C3 6D F4 C3 DA F3 C3 BE : 35
F220 F3 C3 35 F5 C3 9A F9 C3 : F9
F228 76 F3 C3 51 F3 C3 89 F2 : AE
F230 00 00 00 00 00 00 00 00 : 00
F238 00 00 00 00 00 00 00 00 : 00
F240 00 00 00 00 00 00 00 00 : 00
F248 00 00 00 00 00 00 00 00 : 00
F250 00 00 00 00 00 00 00 00 : 00
F258 00 00 00 00 00 00 00 00 : 00
F260 00 00 00 00 00 00 00 00 : 00
F268 00 00 00 00 14 01 00 50 : 65
F270 00 00 00 00 00 03 00 00 : 03
F278 00 00 00 00 00 00 00 00 : 00
SUM: 3B B0 4C 2F 56 A4 03 B2 AA9B

F280 00 00 00 00 00 00 00 00 : 00
F288 00 AF 32 87 F2 3E 10 32 : DA
F290 88 F2 3A 97 E0 32 7D F2 : CC
F298 3E 07 32 8E D1 32 8C D1 : 62
F2A0 2A 78 E0 22 7F F2 2A 7A : B9
F2A8 E0 22 81 F2 3E 03 32 83 : 6B
F2B0 F2 3E 02 32 84 F2 3E 01 : 19
F2B8 32 85 F2 3E 03 32 86 F2 : 94
F2C0 11 83 F2 06 04 CD 18 F2 : 67
F2C8 3A 8C E0 B7 C0 3A 97 E0 : CE
F2D0 32 7E F2 CD 0B F3 CD 22 : 5C
F2D8 F3 3A 88 F2 32 E5 EC AF : 59
F2E0 32 9C E0 CD 0C D0 3A 8C : 1D
F2E8 E0 B7 C0 3A 9C E0 B7 20 : E4
F2F0 F2 CD 39 F3 3A 81 D0 32 : A8
F2F8 82 D0 3A 87 F2 B7 28 03 : E7
SUM: EA BC 52 2A BC 82 8A 69 E85C

F300 32 82 D0 3E 02 32 80 D1 : 47
F308 C3 15 D0 3A 7D F2 E6 01 : 38
F310 C0 3A 7E F2 E6 01 C8 3E : 57
F318 04 32 87 F2 3E 12 32 88 : B9
F320 F2 C9 3A 7D F2 E6 01 C8 : 13
F328 3A 7E F2 E6 01 C0 3E 20 : AF
F330 32 87 F2 3E 13 32 88 F2 : A8
F338 C9 3A E5 EC B7 C0 3A 87 : 0C
F340 F2 FE 20 C8 2A 7F F2 22 : 95
F348 86 D1 2A 81 F2 22 88 D1 : 6F
F350 C9 CD 03 E0 CD 12 D0 CD : F5
F358 15 F2 3A 69 F2 FE 80 D0 : EA
F360 11 54 E0 21 47 F2 01 D0 : AD
F368 00 ED B0 21 55 F2 01 03 : 09
F370 00 ED B0 C3 06 D0 3A 6E : DE
F378 F2 C6 04 5F 16 00 3A 76 : E1
SUM: 39 8D 73 DF F3 34 A1 7D E4A1

F380 F2 21 75 F2 86 21 6C F2 : 7F
F388 86 3C 6F 26 00 CD 68 FA : 86
F390 C5 06 08 21 48 FD 7E 23 : DA
F398 B7 20 06 10 F9 16 20 18 : 34

F3A0 02 16 4D 7A 32 84 D1 C1 : 27
F3A8 3E 20 80 47 3E 03 ED 79 : CC
F3B0 3E 18 80 47 AF ED 79 78 : AA
F3B8 D6 08 47 ED 51 C9 11 49 : 86
F3C0 FD 21 48 FD 01 37 01 AF : 4B
F3C8 77 ED B0 C9 11 49 FD 21 : 55
F3D0 48 FD 01 0F 00 AF 77 ED : 68
F3D8 B0 C9 DD 21 80 EE 3E 01 : 24
F3E0 DD 77 00 DD 77 01 DD 77 : FD
F3E8 03 AF DD 77 02 11 84 EE : 8B
F3F0 21 D8 FD CD 63 F4 21 3C : 77
F3F8 FE CD 63 F4 CD 58 F4 3A : 75
SUM: B3 78 99 49 72 B9 E3 BB 9261

F400 8C E0 B7 C2 CC F3 CD 2A : 9B
F408 D0 3A 84 D1 32 6B F2 2A : 18
F410 91 E0 22 78 F2 CD 15 F2 : D1
F418 CD 48 F4 3A 69 F2 FE 80 : 1C
F420 D0 DD 21 80 EE 3E 01 DD : 58
F428 77 00 DD 77 01 DD 77 02 : 22
F430 AF DD 77 03 11 84 EE 21 : AA
F438 D8 FD CD 63 F4 21 F2 : 53
F440 01 11 00 CD C3 58 FA : BE
F448 3A 84 D1 21 55 FE BE C8 : 89
F450 3E 80 32 69 F2 C3 CC F3 : CD
F458 EB 11 80 EE B7 ED 52 45 : A5
F460 C3 18 F2 7E B7 C8 4F 06 : 1F
F468 00 23 ED B0 C9 1A 13 05 : BB
F470 FE 02 28 25 FE 03 28 2A : A0
F478 CD C2 F4 CD D3 F4 C5 D5 : B1
SUM: 7A 1E 11 27 4C 27 02 B6 8AC6

F480 CD 21 D0 D1 C1 3A 8C E0 : F6
F488 B7 C0 68 3A 84 D1 47 7D : 32
F490 90 6F 26 00 19 EB C3 24 : 10
F498 D0 1A 13 05 32 80 D1 C3 : 48
F4A0 21 D0 CD C2 F4 CD E6 FA : 1B
F4A8 CD D3 F4 3A 58 FD 47 11 : 7B
F4B0 59 FD CD 21 D0 CD F6 FA : CB
F4B8 3A 8C E0 B7 C0 06 00 C3 : E6
F4C0 24 D0 C5 3A 8B E0 4F 06 : B3
F4C8 00 21 C0 EC 09 7E 32 97 : 1D
F4D0 E0 C1 C9 1A 32 80 D1 13 : 1A
F4D8 05 1A 32 65 E0 13 05 1A : C8
F4E0 32 81 D1 13 05 C9 C5 D5 : FF
F4E8 21 68 E0 11 DD EE 01 23 : 69
F4F0 00 ED B0 D1 C1 C9 C5 D5 : 92
F4F8 11 68 E0 21 DD EE 01 23 : 69
SUM: D2 A0 A0 9F 92 72 6D BA 77AE

F500 00 ED B0 D1 C1 C9 11 69 : 72
F508 F2 1A FE 01 20 09 32 6A : D0
F510 F2 2A 78 F2 22 C5 FA CD : 34
F518 A8 F9 A6 20 0B 1A 21 55 : 02
F520 FE BE 30 0C 3C 12 18 E1 : 3F
F528 2F A6 77 CD C9 F2 B7 C9 : 94
F530 3E 80 12 37 C9 CD 80 F9 : 16
F538 30 0E 11 D8 FD 21 58 FD : 9A

F540 01 80 00 ED B0 CD 9A F9 : 7E
F548 3E 01 32 69 F2 3A 55 FE : 59
F550 47 C5 CD 09 F2 3A 73 F2 : 73
F558 FE 04 28 08 11 69 F2 CD : 6B
F560 A8 F9 B6 77 21 69 F2 34 : 7E
F568 C1 10 E6 C9 3A 73 F2 FE : 1D
F570 04 C8 CD 80 F9 30 0E 11 : 61
F578 D8 FD 21 58 FD 01 80 00 : CC
SUM: F0 34 47 4B 0F 5A CB 8E C8CC

F580 ED B0 CD 9A F9 11 69 F2 : 69
F588 CD A8 F9 57 A6 20 04 7A : 09
F590 B6 77 C9 2F A6 77 C9 D5 : E0
F598 DD E1 3A 6E F2 DD 86 00 : BB
F5A0 1F 16 00 3A 76 F2 DD 86 : 7A
F5A8 01 6F 26 00 CD 68 FA ED : B2
F5B0 43 BC FA DD 7E 02 3A BE : 46
F5B8 FA DD 7E 03 32 BF FA ED : 30
F5C0 5B DF EC AF 21 00 38 3E : 6C
F5C8 20 21 20 30 CD D8 F5 3E : 69
F5D0 07 21 0F 20 CD D8 F5 C9 : BA
F5D8 C5 32 C2 FA 78 84 47 32 : 28
F5E0 C1 FA CD 2B F6 CD EA F5 : 55
F5E8 C1 C9 03 DD 6E 02 ED 78 : 3F
F5F0 CD 18 E0 13 3A C2 FA ED : BB
F5F8 79 03 2D 20 F1 ED 4B BC : AE
SUM: F9 FF 21 DC EC 52 44 EC E028

F600 FA 3A C1 FA 47 DD 7E 02 : 93
F608 6F 26 00 09 44 4D DD 6E : 7A
F610 03 E5 3A 6F F2 6F 26 00 : 18
F618 09 44 4D E1 2D C8 ED 78 : D5
F620 CD 18 E0 13 3A C2 FA ED : BB
F628 79 18 E6 7D 32 C0 FA DD : BD
F630 6E 03 DD 66 02 C5 ED 78 : E0
F638 CD 18 E0 13 3A C0 FA ED : B9
F640 79 03 25 20 F1 C1 E5 3A : 92
F648 6F F2 6F 26 00 09 44 4D : 90
F650 E1 2D 20 DE C9 ED 4B BC : C9
F658 FA ED 5B DF EC 3E 38 3E : C1
F660 30 CD 6A F6 3E 20 CD 6A : F2
F668 F6 C9 C5 80 47 32 C1 FA : 38
F670 CD B1 F6 CD 78 F6 C1 C9 : 39
F678 03 3A BE FA 6F CD 15 E0 : 26
SUM: AF 64 BD 9C 64 72 59 A5 CB74

F680 13 ED 79 03 2D 20 F6 ED : AC
F688 4B BC FA 3A C1 FA 47 3A : 77
F690 BE FA 6F 26 00 09 44 4D : E7
F698 3A BF FA 6F E5 3A 6F F2 : E2
F6A0 6F 26 00 09 44 4D E1 2D : 3D
F6A8 C8 CD 15 00 13 ED 79 18 : 1B
F6B0 EB 3A BF FA 6F 3A BE FA : 3F
F6B8 67 C5 CD 15 E0 13 ED 79 : 67
F6C0 03 25 20 F6 C1 E5 3A 6F : 8D
F6C8 F2 6F 26 00 09 44 4D E1 : 02
F6D0 2D 20 E2 C9 3A 68 F2 4F : 16
F6D8 3A 6C F2 81 4F 3A 6B F2 : FF

▶シムシティーの調査で線路を調べると人が住んでいた。公園にも住んでいるし……、まっ
たく、お役所はおにきてんでしょ。

池田 孝志(16)奈良県

F6E0 B9 30 01 4F 3A 68 F2 47 : 14
F6E8 11 70 F2 79 B8 D8 78 12 : 06
F6F0 13 AF 12 1B CD A0 F7 04 : 57
F6F8 18 F1 D5 3A 68 F2 5F 3A : 0B
SUM: 30 B4 71 27 F3 81 99 46 7891
F700 7C F2 BB 38 14 57 3A 6C : 72
F708 F2 83 BA 38 0C 7A 11 70 : 6E
F710 F2 12 13 AF 12 1B CD A0 : 60
F718 F7 3A 69 F2 11 70 F2 12 : 11
F720 13 3E 08 12 1B CD A0 F7 : EA
F728 3A 6D F2 57 3A 69 F2 92 : 17
F730 ED 5B 7A F2 12 D1 C9 C5 : 25
F738 D5 1A FE 80 D2 72 F7 4F : F7
F740 3A 69 F2 81 32 69 F2 4F : F2
F748 3A 6B F2 B9 30 03 32 69 : 1E
F750 F2 3A 69 F2 4F 3A 68 F2 : 6A
F758 47 3A 6C F2 80 B9 D2 9D : 87
F760 F7 3A 6C F2 47 3A 69 F2 : 6B
F768 90 32 68 F2 CD 00 F2 C3 : 9E
F770 9D F7 3A 6D F2 47 3A 69 : 17
F778 F2 4F 1A 81 FE 80 30 08 : 92
SUM: 29 DB 44 DC B1 35 7F 98 9692

F780 B8 38 05 32 69 F2 18 04 : 9E
F788 78 32 69 F2 3A 68 F2 4F : E8
F790 3A 69 F2 B9 D2 9D F7 32 : E6
F798 68 F2 CD 00 F2 D1 C1 C9 : 74
F7A0 CD E6 F8 CD C0 F7 C9 D5 : CD
F7A8 11 70 F2 3A 69 F2 12 CD : E7
F7B0 E6 F8 3A 69 F2 B7 28 06 : 58
F7B8 11 54 F2 3E 2E 12 D1 C9 : 6F
F7C0 C5 D5 E5 3A 68 F2 4F 1A : 7C
F7C8 91 6F 3A 75 F2 85 6F 3A : CF
F7D0 76 F2 85 6F 26 00 13 1A : AF
F7D8 32 71 F2 3A 6E F2 5F 16 : A4
F7E0 00 CD 68 FA CD 89 FA DD : 5C
F7E8 21 B7 FA 3E 28 DD 77 00 : 8C
F7F0 CD 0A F8 21 40 F2 DD 70 : 6F
F7F8 02 CD 4F F8 DD 46 02 23 : 5E
SUM: 95 69 82 34 B0 81 16 B3 B71F

F800 03 DD 35 00 20 F0 E1 D1 : D7
F808 C1 C9 AF DD 77 03 DD 77 : E4
F810 04 3A 71 F2 F6 07 DD 77 : F2
F818 01 E6 08 C0 3A 73 F2 FE : 4C
F820 04 3E 06 CA 4B F8 3A 74 : 03
F828 F2 B7 3E 03 C2 4B F8 3A : 29
F830 72 F2 B7 28 05 16 04 DD : 3F
F838 72 03 3A 73 F2 B7 C8 1E : B1
F840 05 FE 03 28 02 1E 04 DD : 2F
F848 73 04 C9 DD 77 01 C9 3A : 98
F850 B6 FA B7 28 10 F3 3E 1D : ED
F858 D3 00 7E CD 99 30 30 29 : 40
F860 3E 1E D3 00 FB DD 7E 00 : 85
F868 DD 5E 01 DD 56 03 FE 13 : 83

F870 CC D7 F8 FE 12 CC D7 F8 : 46
F878 FE 11 CC D7 F8 FE 08 CC : 7C
SUM: 89 10 2B A3 48 69 21 9A 9ABE
F880 DD F8 7B 5E 16 00 C3 AF : 36
F888 F8 56 23 5E 2B CD 81 2F : 77
F890 CD B6 2F DD B6 01 CD AF : C2
F898 F8 DD 46 02 03 DD 70 02 : 6F
F8A0 23 DD 35 00 CB F2 CD AF : 6E
F8A8 F8 3E 1E D3 00 FB C9 F5 : E0
F8B0 F5 DD DD 7E 02 C6 20 47 : 70
F8B8 ED 79 DD 7E 02 C6 38 47 : 08
F8C0 ED 51 DD 7E 02 C6 30 47 : D8
F8C8 ED 59 F1 C9 00 00 00 00 : 00
F8D0 00 00 00 00 00 00 00 7A : 7A
F8D8 B7 C8 5A AF C9 DD 56 04 : 88
F8E0 7A B7 C8 5A AF C9 C5 D5 : 65
F8E8 E5 ED 53 C3 FA 21 40 F2 : 35
F8F0 11 41 F2 01 27 00 3E 20 : CA
F8F8 77 ED B0 ED 5B C3 FA 1A : 33
SUM: 0F 96 A6 EF 83 CE 59 31 4551

F900 B7 20 06 CD C5 F9 C3 42 : 6D
F908 F9 FE 01 20 09 2A 78 F2 : B5
F910 22 C5 FA C3 34 F9 5F 2A : 5A
F918 C5 FA 3A 6A F2 FE 01 20 : 74
F920 06 2A 78 F2 22 C5 FA 93 : 0E
F928 28 0A 57 DC DA F9 DA EA : F0
F930 F9 22 C5 FA 11 47 F2 CD : F1
F938 03 FA ED 5B C3 FA 1A 32 : 4E
F940 6A F2 CD 54 F9 CD 0E FA : 4B
F948 ED 5B C3 FA 1A 32 7C F2 : BF
F950 E1 D1 C1 C9 AF 32 74 F2 : 83
F958 3A 60 F2 FE 44 21 CD FA : B6
F960 28 15 CD 80 F9 D8 ED 5B : A3
F968 C3 FA CD A8 F9 A6 C8 21 : BA
F970 D2 FA 3E 01 27 00 3E 20 : B4
F978 41 F2 01 05 00 ED B0 C9 : 9F
SUM: 31 A6 D8 80 E8 4A 97 28 0AAE

F980 21 58 FD 11 D8 FD 06 80 : E2
F988 1A BE 20 06 13 23 10 F8 : 3C
F990 B7 C9 78 FE 04 DC 9A F9 : 69
F998 37 C9 11 49 FD 21 48 FD : BD
F9A0 01 0F 00 AF 77 ED B0 C9 : 9C
F9A8 1A CB 3F CB 3F CB 3F 4F : 87
F9B0 06 00 21 48 FD 09 1A E6 : 75
F9B8 07 0E 80 B7 28 05 CB 09 : 4D
F9C0 3D 18 F8 79 C9 11 47 F2 : D9
F9C8 3E 2E 12 13 12 11 60 F2 : 06
F9D0 3E 44 12 C9 F5 7A ED 44 : FD
F9D8 57 01 17 00 09 CD 12 E0 : 37
F9E0 23 FE 0D 20 F8 15 20 FA : 6F
F9E8 F1 C9 F5 01 19 00 B7 ED : 6D
F9F0 42 CD FA F9 23 15 20 FE : 50
F9F8 F1 C9 CD 12 E0 FE 0D C8 : 4C

SUM: A8 78 82 58 B4 74 76 1C A221
FA00 2B 18 F7 CD 12 E0 FE 0D : 04
FA08 C8 23 12 13 18 F5 AF 32 : FE
FA10 72 F2 32 73 F2 3A 60 F2 : 87
FA18 FE 4A 20 06 3E 04 32 73 : 4F
FA20 F2 C9 21 C7 FA CD 47 FA : AB
FA28 21 CA FA CD 47 FA 16 03 : 0C
FA30 3A 60 F2 FE 4D 28 0B 16 : 20
FA38 01 FE 42 28 05 16 02 FE : 84
FA40 41 C0 7A 32 73 F2 C9 C5 : A0
FA48 11 55 F2 06 03 1A 13 CD : 5B
FA50 5F FA BE 23 20 07 10 F5 : 66
FA58 3E 01 32 72 F2 C1 C9 FE : 5D
FA60 61 D8 FE 7B D8 D6 20 C9 : 41
FA68 3A 6F F2 FE 28 CA 7D FA : 02
FA70 29 29 29 29 E5 29 29 19 : BF
FA78 D1 19 44 4D C9 29 29 29 : F4
SUM: 35 FB 63 CF 1B DE 4D 3F EA72

FA80 E5 29 29 19 D1 19 44 4D : CB
FA88 C9 C5 3A 77 F2 B7 3E 00 : 26
FA90 20 1F 3A 7F D0 B7 28 19 : C0
FA98 F3 3E 1D ED 79 3A 00 00 : EE
FAA0 4F 3A 00 10 47 3E 1E D3 : 0F
FAA8 00 FB 78 B9 3E 01 20 01 : 8C
FAB0 AF 32 B6 FA C1 C9 00 00 : 1B
FAB8 00 00 00 00 00 00 00 00 : 00
FAC0 00 00 00 00 00 00 00 58 : 58
FAC8 31 20 42 41 54 3C 44 49 : F1
FAD0 52 3E 3C 4D 52 4B 3E : F4
SUM: 42 10 66 4D F8 50 6A DB 19A7

リスト6

FE80 00 00 00 00 00 00 00 00 : 00
FE88 00 00 00 00 00 06 00 00 : 06
FE90 00 00 00 00 00 00 00 00 : 00
FE98 00 00 00 00 00 00 04 03 : 07
FEA0 5F 00 00 00 00 00 00 00 : 5F
FEA8 00 00 00 00 00 0A 07 5E : D8
FEB0 00 03 01 00 5C 08 5D 04 : C9
FEB8 02 00 00 00 00 00 00 00 : 02
FEC0 00 5B 5B 5B 5B 5B 5B 00 : 22
FEC8 00 00 00 00 00 00 00 00 : 00
FED0 00 00 00 00 00 00 00 5B : 5B
FED8 5B 5B 00 00 00 00 00 00 : 06
FEE0 00 5B 5B 5B 5B 5B 5B 00 : 22
FEE8 00 00 00 00 00 00 00 00 : 00
FEF0 00 00 00 00 00 00 00 5B : 5B
FEF8 5B 5B 00 00 00 00 00 00 : 06
SUM: 17 6F 17 BF 12 CE 1E 1B 2888

リスト7

0000 1 ;
0000 2 ;MENU.X1
0000 3 ; WITH KAME-DOS
0000 4 ;
0000 5 ; FOR S-O5 REDA Ver 1.0
0000 6 ;
C200 7 ORG \$C200
C208 P 8
C208 P 9 MRKSTK EQU \$CD48
C208 P 10 NMDIR1 EQU \$CD58
C208 P 11 NMDIR2 EQU \$CDD8
C208 P 12 POSCRS EQU \$CE58
C200 13
D080 P 14 \$ZOKU EQU \$D080 ;POKE ADR.
D180 P 15 \$OD EQU \$D180
D181 P 16 \$OP EQU \$D181
D182 P 17 \$FCRS EQU \$D182
D184 P 18 \$YEN EQU \$D184
D186 P 19 \$FSZL EQU \$D186
D18A P 20 \$DDR EQU \$D18A
E054 P 21 \$FLNM EQU \$E054
E055 P 22 \$SDDR EQU \$E055
E058 P 23 \$FNAM EQU \$E058
E078 P 24 \$FBYT EQU \$E078
E08C P 25 \$STOP EQU \$E08C
E08B P 26 \$DN EQU \$E08B
E091 P 27 \$BF EQU \$E091
E097 P 28 \$MAC EQU \$E097
E09C P 29 \$IOFG EQU \$E09C
ECC0 P 30 \$MAC4 EQU \$ECC0
ECC5 P 31 \$ESCP EQU \$ECC0+37
ECDF P 32 \$SBUFF EQU \$ECDF
E880 P 33 P256 EQU \$E800+\$80
C200 34
E003 P 35 \$DLFAT EQU \$E003 ;CALL ADR.
E012 P 36 \$LDAHL EQU \$E012
E015 P 37 \$LDADE EQU \$E015
E018 P 38 \$LDDEA EQU \$E018
D006 P 39 \$DIR2 EQU \$D006
D00C P 40 \$DEVI EQU \$D00C
D012 P 41 \$DLDIR EQU \$D012
D015 P 42 \$SAVED EQU \$D015
D021 P 43 \$PREOP EQU \$D021
D025 P 44 \$OPENS EQU \$D024
D02A P 45 \$DIRSB EQU \$D02A
F281 P 46 \$FTJIS EQU \$F281
F286 P 47 \$JISVR EQU \$F286
3099 P 48 \$FTCHK EQU \$3099
C200 49 ;
C200 C3 DA C6 50 DIR JP DIRS
C203 C3 00 C7 51 CRPRT JP CRPRTS
C206 C3 3D C7 52 LIMIT JP LIMITS ;USR
C209 C3 AD C7 53 FNRWK JP FNRKRS
C20C C3 97 C5 54 WIDOW JP WIDOWS ;
C20F C3 58 C6 55 WIDOW JP WIDOWS
C212 C3 6C C6 56 MARK JP MARKS
C215 C3 06 C5 57 CHECK JP CHECKS
C218 C3 6D C4 58 OPEN JP OPEN1 ;
C21B C3 DA C3 59 TWOSTP JP TWOSTS
C21E C3 BE C3 60 CLRWK JP CLRWKS

C221 C3 35 C5 61 MACON JP ALLONS
C224 C3 95 C9 62 MACOF JP MRKCLR
C227 C3 76 C3 63 MARK? JP MARKS?
C22A C3 51 C3 64 DEL JP DELS
C22D C3 89 C2 65 COPY JP COPIYS
C230 66 ;
C230 67 ;
C240 68 ORG \$C240
C240 69
C240 70 WORK DS 40
C268 00 71 DU DB 0
C269 00 72 CR DB 0
C26A 00 73 CB DB 0
C26B 00 74 DB DB 0
C26C 14 75 YS DB 20
C26D 01 76 UO DB 1
C26E 00 77 XX DB 0
C26F 50 78 WIDTH DB 80
C270 79 WWF DB 5 ;n,ATR
C275 03 80 USP DB 3
C276 00 81 YY DB 0
C277 00 82 KMODE DB 0
C278 00 83 NWADR DB 0
C27A 00 84 CBADR DB 0
C27C 00 85 CSZ DB 0
C27D 00 86
C27D 00 87 MAC1 DB 0
C27E 00 88 MAC2 DB 0
C27F 89 FBKFS DS 4
C283 90 UUSR DS 4
C287 00 91 DSHU DB 0
C288 00 92 ESCW DB 0
C289 93
C289 94 ;
C289 95 ;COPY
C289 96
C289 97 COPIYS
C289 AF 98 XOR A
C28A 32 87 C2 99 LD (DSHU),A
C28D 3E 10 100 LD A,\$10
C28F 32 88 C2 101 LD (ESCW),A
C292 102
C292 3A 97 E0 103 LD A,(\$MAC)
C293 32 7D C2 104 LD (MAC1),A
C298 3E 07 105 LD A,7
C29A 32 8B D1 106 LD (\$DDR+1),A
C29D 32 8C D1 107 LD (\$DDR+2),A
C2A0 2A 78 E0 108 LD HL,(\$FBYT)
C2A3 22 7F C2 109 LD (FBXFS),HL
C2A6 2A 7A E0 110 LD HL,(\$FBYT+2)
C2A9 22 81 C2 111 LD (FBXFS+2),HL
C2AC 3E 03 112 LD A,3
C2AE 32 83 C2 113 LD (UUSR),A
C2B1 3E 02 114 LD A,2
C2B3 32 84 C2 115 LD (UUSR+1),A
C2B6 3E 01 116 LD A,1
C2B8 32 85 C2 117 LD (UUSR+2),A
C2BB 3E 03 118 LD A,3
C2BD 32 86 C2 119 LD (UUSR+3),A
C2C0 11 83 C2 120 LD DE,UUSR

C2C3 06 04 121 LD B,4
C2C5 CD 18 C2 122 CALL OPEN
C2C8 3A 8C E0 123 LD A,(\$STOP)
C2CB B7 124 OR A
C2CC C0 125 RET NZ
C2CD 3A 97 E0 126 LD A,(\$MAC)
C2D0 32 7E C2 127 LD (MAC2),A
C2D3 128
C2D3 CD 0B C3 129 CALL BYMAC
C2D6 CD 22 C3 130 CALL BYMAC2
C2D9 3A 88 C2 131 LD A,(ESCW)
C2DC 32 E5 EC 132 LD (\$ESCP),A
C2DF AF 133 XOR A
C2E0 32 9C E0 134 LD (\$IOFG),A
C2E3 00 135 COPYL
C2E3 CD 0C D0 136 CALL \$DEVI
C2E6 3A 8C E0 137 LD A,(\$STOP)
C2E9 B7 138 OR A
C2EA C0 139 RET NZ
C2EB 3A 9C E0 140 LD A,(\$IOFG)
C2EE B7 141 OR A
C2EF 2D F2 142 JR NZ,COPYL
C2F1 143
C2F1 CD 39 C3 144 CALL CLFSZ
C2F4 3A 81 D0 145 LD A,(\$ZOKU+1)
C2F7 32 82 D0 146 LD (\$ZOKU+2),A
C2FA 3A 87 C2 147 LD A,(DSHU)
C2FD B7 148 OR A
C2FE 28 03 149 JR Z,ZOKUSK
C300 32 82 D0 150 LD (\$ZOKU+2),A
C303 151 ZOKUSK
C303 3E 02 152 LD A,2
C305 32 80 D1 153 LD (\$OD),A
C308 C3 15 D0 154 JP \$SAVED
C30B 155
C30B 156 BYMAC
C30B 3A 7D C2 157 LD A,(\$MAC1)
C30E E6 01 158 AND 1
C310 C0 159 RET NZ
C311 3A 78 C2 160 LD A,(\$MAC2)
C314 E6 01 161 AND 1
C316 C8 162 RET Z
C317 3E 04 163 LD A,4
C319 32 87 C2 164 LD (DSHU),A
C31C 3E 12 165 LD A,\$12
C31F 32 88 C2 166 LD (ESCW),A
C321 C9 167 RET
C322 168
C322 3A 7D C2 169 BYMAC2
C325 E6 01 170 LD A,(\$MAC1)
C327 C8 171 AND 1
C328 3A 78 C2 172 LD A,(\$MAC2)
C32B E6 01 173 AND 1
C32D C0 174 RET NZ
C32E 3E 20 175 LD A,\$20
C330 32 87 C2 176 LD (DSHU),A
C333 3E 13 177 LD A,\$13
C335 32 88 C2 178 LD (ESCW),A
C338 C9 179 RET


```

C339 181
C339 182 CLFSZ
C339 3A E5 EC 183 LD A,(RSCFP)
C33C B7 184 OR A
C33D C0 185 RET NZ
C33E 3A 87 C2 186 LD A,(DSHU)
C341 FE 20 187 CP #20
C343 C8 188 RET Z
C344 2A 7F C2 189 LD HL,(FBXFS)
C347 22 86 D1 190 LD (#FSZL),HL
C34A 2A 81 C2 191 LD HL,(FBXFS+2)
C34D 22 86 D1 192 LD (#FSZL+2),HL
C350 C9 193 RET
C351 194
C351 195
C351 196 ;DEL
C351 197
C351 198 DELS
C351 CD 03 E0 199 CALL #DLFAT
C354 CD 12 D0 200 CALL #DLDIR
C357 CD 15 C2 201 CALL CHECK
C35A 3A 69 C2 202 LD A,(CR)
C35D FE 80 203 CP 128
C35F D0 204 RET NC
C360 11 54 E0 205 LD DE,#FLNM
C363 21 47 C2 206 LD HL,WORK+7
C366 01 0D 00 207 LD BC,13
C369 E0 208 LDIR
C36B 21 55 C2 209 LD HL,WORK+21
C36E 01 03 00 210 LD BC,3
C371 ED 80 211 LDIR
C373 C3 06 D0 212 JP #DIR2
C376 213
C376 ;
C376 ;MARK?
C376 216 ;
C376 217 MARKS?
C376 218 LD A,(XX)
C379 C6 84 219 ADD A,A
C37B 5F 220 LD E,A
C37C 16 00 221 LD D,0
C37E 3A 76 C2 222 LD A,(YY)
C381 21 75 C2 223 LD HL,USP
C384 86 224 ADD A,(HL)
C385 21 6C C2 225 LD HL,YS
C388 86 226 ADD A,(HL)
C389 3C 227 INC A
C38A 6F 228 LD L,A
C38B 26 00 229 LD H,0
C38D CD 63 CA 230 CALL XYADR
C390 231
C390 C5 232 PUSH BC
C391 06 08 233 LD B,8
C393 21 48 CD 234 LD HL,MRKSTK
C396 235 MARKSLP
C396 7E 236 LD A,(HL)
C397 23 237 INC HL
C398 B7 238 OR A
C399 20 06 239 JR NZ,MARKS?
C39B 10 F9 240 DJNZ MARKSLP
C39D 16 20 241 LD D," "
C39F 18 02 242 JR MARKS?
C3A1 243 MARKS?
C3A1 16 4D 244 LD D,"H"
C3A3 245 MARKS?
C3A3 7A 246 LD A,D
C3A4 32 84 D1 247 LD (#YEN),A
C3A7 C1 248 POP BC
C3A8 3E 20 249 LD A,#20
C3AA 80 250 ADD A,B
C3AB 47 251 LD B,A
C3AC 3E 03 252 LD A,3
C3AE ED 79 253 OUT (C),A
C3B0 3E 18 254 LD A,#18
C3B2 86 255 ADD A,B
C3B3 47 256 LD B,A
C3B4 AF 257 XOR A
C3B5 ED 79 258 OUT (C),A
C3B7 78 259 LD A,B
C3B8 D6 08 260 SUB 8
C3BA 47 261 LD B,A
C3BB ED 51 262 OUT (C),D
C3BD C9 263 RET
C3BE 264
C3BE 265 ;
C3BE ;CLEAR WORKS
C3BE 267 ;
C3BE 268 CLRWKS
C3BE 11 49 CD 269 LD DE,MRKSTK+1
C3C1 21 48 CD 270 LD HL,MRKSTK
C3C4 01 37 01 271 LD BC,15+256+40
C3C7 AF 272 XOR A
C3C8 77 273 LD (HL),A
C3C9 ED B0 274 LDIR
C3CB C9 275 RET
C3CC 276
C3CC CLRMRK
C3CC 11 49 CD 278 LD DE,MRKSTK+1
C3CF 21 48 CD 279 LD HL,MRKSTK
C3D2 01 0F 00 280 LD BC,15
C3D5 AF 281 XOR A
C3D6 77 282 LD (HL),A
C3D7 ED B0 283 LDIR
C3D9 C9 284 RET
C3DA 285
C3DA 286 ;
C3DA ;OPEN#2
C3DA 288 ;
C3DA 289 TMOSTS
C3DA DD 21 80 EE 290 LD IX,P256
C3DE 3E 01 291 LD A,1
C3E0 DD 77 00 292 LD (IX+0),A
C3E3 DD 77 01 293 LD (IX+1),A
C3E6 DD 77 03 294 LD (IX+3),A
C3E9 AF 295 XOR A
C3EA DD 77 02 296 LD (IX+2),A
C3ED 11 84 EE 297 LD DE,P256+4
C3F0 21 D8 CD 298 LD HL,NMIDIR2
C3F3 CD 63 C4 299 CALL PDEHL
C3F6 21 3C CE 300 LD HL,NMIDIR2+100
C3F9 CD 63 C4 301 CALL PDEHL
C3FC CD 58 C4 302 CALL TWOPN
C3FF 3A 8C E0 303 LD A,(#STOP)
C402 B7 304 OR A
C403 C2 CC C3 305 JP NZ,CLMRK
C406 306
C406 CD 2A D0 307 CALL #DIRSB
C409 3A 84 D1 308 LD A,(#YEN)
C40C 32 6B C2 309 LD (DE),A
C40F 2A 91 E0 310 LD HL,(#BF)
C412 22 78 C2 311 LD (NWADR),HL
C415 CD 15 C2 312 CALL CHECK
C418 CD 46 C4 313 CALL DSKIDF
C41B 3A 69 C2 314 LD A,(CR)
C41E FE 80 315 CP 128
C420 D0 316 RET NC
C421 DD 21 80 EE 317 LD IX,P256
C425 3E 01 318 LD A,1
C427 DD 77 01 319 LD (IX+0),A
C42A DD 77 01 320 LD (IX+1),A

```

```

C42D DD 77 02 321 LD (IX+2),A
C430 AF 322 XOR A
C431 DD 77 03 323 LD (IX+3),A
C434 11 84 EE 324 LD DE,P256+4
C437 21 D8 CD 325 LD HL,NMIDIR2
C43A CD 63 C4 326 CALL PDEHL
C43D 21 47 C2 327 LD HL,WORK+7
C440 01 11 00 328 LD BC,17
C443 ED B0 329 LDIR
C445 C3 58 C4 330 JP TWOPN
C448 331
C448 DSKIDF
C448 332
C448 LD A,(#YEN)
C44B 21 55 CE 334 LD HL,NMIDIR2+125
C44E BE 335 CP (HL)
C44F C8 336 RET Z
C450 3E 80 337 LD A,128
C452 32 69 C2 338 LD (CR),A
C455 C3 CC C3 339 JP CLMRK
C458 340
C458 341
C458 TWOPN
C458 342
C458 EX DE,HL
C45B EB 344 LD DE,P256
C45C B7 345 OR A
C45D ED 52 346 SBC HL,DE
C45F 45 347 LD B,L
C460 C3 18 C2 348 JP OPEN
C463 349
C463 PDEHL
C463 7E 351 LD A,(HL)
C464 B7 352 OR A
C465 C8 353 RET Z
C466 4F 354 LD C,A
C467 06 00 355 LD B,0
C469 23 356 INC HL
C46A ED B0 357 LDIR
C46C C9 358 RET
C46D 359
C46D 360 ;
C46D ;"OPEN"
C46D 362 ;
C46D 363 OPEN1
C46D 1A 364 LD A,(DE)
C46E 13 365 INC DE
C46F 05 366 DEC B
C470 FE 02 367 CP 2
C472 28 25 368 JR Z,OPEN2
C474 FE 83 369 CP 3
C476 28 2A 370 JR Z,OPEN3
C478 371
C478 CD C2 C4 372 CALL SMACDN
C47B CD D3 C4 373 CALL ODSBOP
C47E D5 374 PUSH BC
C47F D5 375 PUSH DE
C480 CD 21 D0 376 CALL #PREOP
C483 D1 377 POP DE
C484 C1 378 POP BC
C485 3A 8C E0 379 LD A,(#STOP)
C48B 07 380 OR A
C489 C0 381 RET NZ
C48A 68 382 LD L,B
C48B 3A 84 D1 383 LD A,(#YEN)
C48E 47 384 LD B,A
C48F 7D 385 LD L,L
C490 90 386 SUB B
C491 6F 387 LD L,A
C492 26 00 388 LD H,0
C494 19 389 ADD HL,DE
C495 EB 390 EX DE,HL
C496 C3 24 D0 391 JP #OPENS
C499 392
C499 OPEN2
C499 1A 394 LD A,(DE)
C49A 13 395 INC DE
C49B 05 396 DEC B
C49C 32 80 D1 397 LD (#OD),A
C49F C3 21 D0 398 JP #PREOP
C4A2 399
C4A2 OPEN3
C4A2 CD C2 C4 400 CALL SMACDN
C4A5 CD E6 C4 402 CALL STKFN
C4A8 CD D3 C4 403 CALL ODSBOP
C4AB 3A 58 CD 404 LD A,(NMIDIR1)
C4AE 47 405 LD B,A
C4AF 11 59 CD 406 LD DE,NMIDIR1+1
C4B2 CD 21 D0 407 CALL #PREOP
C4B5 CD F6 C4 408 CALL LODFN
C4B8 3A 8C E0 409 LD A,(#STOP)
C4BB B7 410 OR A
C4BC C0 411 RET NZ
C4BD 06 00 412 LD B,0
C4BF C3 24 D0 413 JP #OPENS
C4C2 414
C4C2 415 ;
C4C2 416 SMACDN
C4C2 C5 417 PUSH BC
C4C3 3A 8B E0 418 LD A,(#DN)
C4C6 4F 419 LD C,A
C4C7 06 00 420 LD B,0
C4C9 21 C0 CE 421 LD HL,AMAC4
C4CC 09 422 ADD HL,BC
C4CD 7E 423 LD A,(HL)
C4CE 32 97 E0 424 LD (#MAC),A
C4D1 C1 425 POP BC
C4D2 C9 426 RET
C4D3 427
C4D3 ODSBOP
C4D3 1A 429 LD A,(DE)
C4D4 32 80 D1 430 LD (#OD),A
C4D7 13 431 INC DE
C4D8 05 432 DEC B
C4D9 1A 433 LD A,(DE)
C4DA 32 65 E0 434 LD (#SDR),A
C4DD 13 435 INC DE
C4DE 05 436 DEC B
C4DF 1A 437 LD A,(DE)
C4E0 32 81 D1 438 LD (#OP),A
C4E3 13 439 INC DE
C4E4 05 440 DEC B
C4E5 C9 441 RET
C4E6 442
C4E6 STKFN
C4E7 D5 444 PUSH BC
C4E8 21 68 E0 446 LD HL,#FNAM
C4EB 11 DD EE 447 LD DE,P256+5D
C4EE 01 23 00 448 LD BC,35
C4F1 ED B0 449 LDIR
C4F3 D1 450 POP DE
C4F4 C1 451 POP BC
C4F5 C9 452 RET
C4F6 453
C4F6 LODFN
C4F6 C5 458 PUSH BC
C4F7 D5 456 PUSH DE
C4F8 11 68 E0 457 LD DE,#FNAM
C4FB 21 DD EE 458 LD HL,P256+5D
C4FE 01 23 00 459 LD BC,35
C501 ED B0 460 LDIR

```

```

C503 D1 461 POP DE
C504 C1 462 POP BC
C505 C9 463 RET
C506 464
C506 465 ;
C506 466 ;MARK CHECK
C506 467
C506 468 CHECKS ;CF
C506 11 69 C2 469 LD DE,CR
C509 470
C509 471 LD A,(DE)
C50A FE 01 472 CP 1
C50C 20 09 473 JR NZ,CHKLPK
C50E 32 6A C2 474 LD (C8),A
C511 2A 78 C2 475 LD HL,(NWADR)
C514 22 08 CA 476 LD (WTOA),HL
C517 477
C517 CD A3 C9 478 CALL RRCA
C51A A6 479 AND (HL)
C51B 20 0B 480 JR NZ,CHKS1
C51D 1A 481 LD A,(DE)
C51E 21 55 CE 482 LD HL,NMIDIR2+125
C521 BE 483 CP (HL)
C522 30 0C 484 JR NC,CHKS2
C524 3C 485 INC A
C525 12 486 LD (DE),A
C526 18 E1 487 JR CHKL
C528 488
C528 489
C529 A6 490 AND (HL)
C52A 77 491 LD (HL),A
C52B CD 09 C2 492 CALL FNWRK
C52E B7 493 OR A
C52F C9 494 RET
C530 495
C530 496 CHKS2
C530 3E 80 497 LD A,128
C532 12 498 LD (DE),A
C533 37 499 SCF
C534 C9 500 RET
C535 501
C535 502 ;
C535 503 ;ALL MARK
C535 504
C535 505 ALLONS
C535 CD 7B C9 506 CALL DIRIDF
C538 30 0E 507 JR NC,ALLSSB
C53A 508
C53A 11 D8 CD 509 LD DE,NMIDIR1
C53D 21 58 CD 510 LD HL,NMIDIR1
C540 01 80 00 511 LD BC,128
C543 ED B0 512 LDIR
C545 CD 95 C9 513 CALL MRKCLR
C548 514
C548 515
C548 3E 01 516 LD A,1
C54A 32 69 C2 517 LD (CR),A
C54D 3A 55 CE 518 LD A,(NMIDIR2+125)
C550 47 519 LD B,A
C551 520
C551 521 PUSH BC
C552 CD 09 C2 522 CALL FNWRK
C555 3A 73 C2 523 LD A,(WWW+3)
C558 FE 04 524 CP 4
C55A 28 08 7F 525 JR Z,ALLSK
C55C 11 69 C2 526 LD DE,CR
C55F CD A3 C9 527 CALL RRCA
C562 B6 528 OR (HL)
C563 77 529 LD (HL),A
C564 530
C564 21 69 C2 531 LD HL,CR
C567 34 532 INC (HL)
C568 C1 533 POP BC
C569 10 E6 534 DJNZ ALLLP
C56B C9 535 RET
C56C 536
C56C 537 ;
C56C 538 ;MARK
C56C 539 ;
C56C 540 MARKS
C56C 3A 73 C2 541 LD A,(WWW+3)
C56F FE 04 542 CP 4
C571 C8 543 RET Z
C572 CD 7B C9 544 CALL DIRIDF
C575 30 0E 545 JR NC,MARKSSB
C577 546
C577 11 D8 CD 547 LD DE,NMIDIR2
C57A 21 58 CD 548 LD HL,NMIDIR1
C57D 01 80 00 549 LD BC,128
C580 ED B0 550 LDIR
C582 CD 95 C9 551 CALL MRKCLR
C585 552
C585 553 MARKSSB
C585 11 69 C2 554 LD DE,CR
C588 CD A3 C9 555 CALL RRCA
C58B 57 556 LD D,A
C58C A6 557 AND (HL)
C58D 20 04 558 JR NZ,MARKSSK
C58F 7A 559 LD A,D
C590 B6 560 OR (HL)
C591 77 561 LD (HL),A
C592 C9 562 RET
C593 563
C593 564 CPL
C594 A6 565 AND (HL)
C595 77 566 LD (HL),A
C596 C9 567 RET
C597 568
C597 569 ;
C597 570 ;WINDOW
C597 571 ;
C597 572 WIDOWS
C597 D5 573 PUSH DE
C598 DD E1 574 POP IX
C59A 3A 6E C2 575 LD A,(XX)
C59D DD 86 00 576 ADD A,(IX+0)
C5A0 5F 577 LD E,A
C5A1 16 00 578 LD D,0
C5A3 3A 76 C2 579 LD A,(YY)
C5A6 DD 86 01 580 ADD A,(IX+1)
C5A9 6F 581 LD L,A
C5AA 26 00 582 LD H,0
C5AC CD 63 CA 583 CALL XYADR
C5AF ED 43 B7 CA 584 LD (WYSS),BC
C5B3 DD 7E 02 585 LD A,(IX+2)
C5B6 32 B9 CA 586 LD (WYSS+2),A
C5B9 DD 7E 03 587 LD A,(IX+3)
C5BC 32 BA CA 588 LD (WYSS+3),A
C5BF ED 5B DF EC 589 LD DE,(#5BUFF)
C5C3 AF 590 XOR A
C5C4 21 00 38 591 LD HL,#3800
C5C7 CD DB C5 592 CALL WNSB ;X1ではこの
C5CA 593 ;3バイトなし
C5CA 3E 20 594 LD A,#20
C5CC 21 20 30 595 LD HL,#3020
C5CF CD DB C5 596 CALL WNSB
C602 3E 07 597 LD A,7
C604 21 0F 20 598 LD HL,#200F
C607 CD DB C5 599 CALL WNSB
C60A 600 RET

```

▶学生のころはPC-9801が買いたかった。会社でFM-Rを使い出した。なにかが、なにかが違う！ て、X68000になったのである。セグメントのない世界はいないな。

和田 岳雄(21)福岡県

```

C5DB      601 WND5B
C5DB      602 WND5B
C5DB C5    603 PUSH BC
C5DC 32 BD CA 604 LD A,(WXYSS+6),A
C5DF 78    605 LD A,B
C5E0 84    606 ADD A,H
C5E1 47    607 LD B,A
C5E2 32 BC CA 608 LD (WXYSS+5),A
C5E5 CD 2E C6 609 CALL WND5B2
C5E8 CD ED C5 610 CALL KAGE
C5EB C1     611 POP BC
C5EC C9     612 RET
C5ED        613
C5ED        614 KAGE
C5ED 03     615 INC BC
C5EE DD 6E 02 616 LD L,(IX+2)
C5F1        617 KAGELPY
C5F1 ED 78 618 IN A,(C)
C5F3 CD 18 E0 619 CALL #LDDEA
C5F6 13     620 INC DE
C5F7 3A BD CA 621 LD A,(WXYSS+6)
C5FA ED 79 622 OUT (C),A
C5FC 03     623 INC BC
C5FD 2D     624 DEC L
C5FE 20 F1 625 JR NZ,KAGELPY
C600 ED 4B B7 CA 626 LD BC,(WXYSS)
C604 3A BC CA 627 LD A,(WXYSS+5)
C607 47     628 LD B,A
C608 DD 7E 02 629 LD A,(IX+2)
C60B 6F     630 LD L,A
C60C 26 00 631 LD H,0
C60E 09     632 ADD HL,BC
C60F 44     633 LD B,H
C610 4D     634 LD C,L
C611 DD 6E 03 635 LD L,(IX+3)
C614        636 KAGELPY
C614 E5     637 PUSH HL
C615 3A 6F C2 638 LD A,(WIDTH)
C618 6F     639 LD L,A
C619 26 00 640 LD H,0
C61B 09     641 ADD HL,BC
C61C 44     642 LD B,H
C61D 4D     643 LD C,L
C61E E1     644 POP HL
C61F 2D     645 DEC L
C620 C8     646 RET Z
C621 ED 78 647 IN A,(C)
C623 CD 18 E0 648 CALL #LDDEA
C626 13     649 INC DE
C627 3A BD CA 650 LD A,(WXYSS+6)
C62A ED 79 651 OUT (C),A
C62C 18 E6 652 JR KAGELPT
C62E        653
C62E        654 WND5B2
C62E 7D     655 LD A,L
C62F 32 BB CA 656 LD (WXYSS+4),A
C632 DD 6E 03 657 LD L,(IX+3)
C635        658 WNDLP1
C635 DD 66 02 659 LD H,(IX+2)
C638 C5     660 PUSH BC
C639        661 WNDLP2
C639 ED 78 662 IN A,(C)
C63B CD 18 E0 663 CALL #LDDEA
C63E 13     664 INC DE
C63F 3A BB CA 665 LD A,(WXYSS+4)
C642 ED 79 666 OUT (C),A
C644 03     667 INC BC
C645 25     668 DEC H
C646 20 F1 669 JR NZ,WNDLP2
C648 C1     670 POP BC
C649 E5     671 PUSH HL
C64A 3A 6F C2 672 LD A,(WIDTH)
C64D 5F     673 LD L,A
C64E 26 00 674 LD H,0
C650 09     675 ADD HL,BC
C651 44     676 LD B,H
C652 4D     677 LD C,L
C653 E1     678 POP HL
C654 2D     679 DEC L
C655 20 DE 680 JR NZ,WNDLP1
C657 C9     681 RET
C658        682
C658        683 ;
C658        684 ; WINDOW BACK
C658        685 ;
C658        686 UNDO5
C658 ED 4B B7 CA 687 LD BC,(WXYSS)
C65C ED 5B DF EC 688 LD DE,(#5BUFF)
C660 3E 38 689 LD A,#38
C662 CD 70 C6 690 CALL UNDSBB ;X1ではこの
C665        691 ;3バイトなし
C665 3E 30 692 LD A,#30
C667 CD 70 C6 693 CALL UNDSBB
C66A 3E 20 694 LD A,#20
C66C CD 70 C6 695 CALL UNDSBB
C66F C9     696 RET
C670        697
C670        698 UNDSBB
C670 C5     699 PUSH BC
C671 80     700 ADD A,B
C672 47     701 LD B,A
C673 32 BC CA 702 LD (WXYSS+5),A
C676 CD B7 C6 703 CALL UNDSBB2
C679 CD 7E C6 704 CALL UNKAGE
C67C C1     705 POP BC
C67D C6     706 RET
C67E        707
C67E        708 UNKAGE
C67E 03     709 INC BC
C67F 3A B9 CA 710 LD A,(WXYSS+2)
C682 5F     711 LD L,A
C683        712 UNKGLPY
C683 CD 15 E0 713 CALL #LDADE
C686 13     714 INC DE
C687 ED 79 715 OUT (C),A
C689 03     716 INC BC
C68A 2D     717 DEC L
C68B 20 F6 718 JR NZ,UNKGLPY
C68D ED 4B B7 CA 719 LD BC,(WXYSS)
C691 3A BC CA 720 LD A,(WXYSS+5)
C694 47     721 LD B,A
C695 3A B9 CA 722 LD (WXYSS+2)
C698 6F     723 LD L,A
C699 26 00 724 LD H,0
C69B 09     725 ADD HL,BC
C69C 44     726 LD B,H
C69D 4D     727 LD C,L
C69E 3A BA CA 728 LD A,(WXYSS+3)
C6A1 6F     729 LD L,A
C6A2        730 UNKGLPT
C6A2 5F     731 PUSH HL
C6A3 3A 6F C2 732 LD A,(WIDTH)
C6A6 6F     733 LD L,A
C6A7 26 00 734 LD H,0
C6A9 09     735 ADD HL,BC
C6AA 44     736 LD B,H
C6AB 4D     737 LD C,L
C6AC E1     738 POP HL
C6AD 2D     739 DEC L
C6AE C8     740 RET Z

```

```

C6AF CD 15 E0 741 CALL #LDADE
C6B2 13     742 INC DE
C6B3 ED 79 743 OUT (C),A
C6B5 18 EB 744 JR UNKGLPT
C6B7        745
C6B7        746 UNDSBB2
C6B7 3A BA CA 747 LD A,(WXYSS+3)
C6BA 6F     748 LD L,A
C6BB        749 UNDLPI
C6BB 3A B9 CA 750 LD A,(WXYSS+2)
C6BE 67     751 LD H,A
C6BF C5     752 PUSH BC
C6C0        753 UNDLPI
C6C0 CD 15 E0 754 CALL #LDADE
C6C3 13     755 INC DE
C6C4 ED 79 756 OUT (C),A
C6C6 03     757 INC BC
C6C7 25     758 DEC H
C6C8 20 F6 759 JR NZ,UNDLPI
C6CA C1     760 POP BC
C6CB E5     761 PUSH HL
C6CC 3A 6F C2 762 LD A,(WIDTH)
C6CF 6F     763 LD L,A
C6D0 26 00 764 LD H,0
C6D2 09     765 ADD HL,BC
C6D3 44     766 LD B,H
C6D4 4D     767 LD C,L
C6D5 E1     768 POP HL
C6D6 2D     769 DEC L
C6D7 20 E2 770 JR NZ,UNDLPI
C6D9 C9     771 RET
C6DA        772
C6DA        773 ;
C6DA        774 ; FOR DU TO K
C6DA        775 ;
C6DA        776 DIRS
C6DA 3A 68 C2 777 LD A,(DU)
C6DD 4F     778 LD C,A
C6DE 3A 6C C2 779 LD A,(YS)
C6E1 81     780 ADD A,C
C6E2 4F     781 LD C,A
C6E3 3A 68 C2 782 LD A,(DDE)
C6E6 B9     783 CP C
C6E7 30 01 784 JR NC,DIRSSK
C6E9 4F     785 LD C,A
C6EA        786 DIRSSK
C6EA 3A 68 C2 787 LD A,(DU)
C6ED 47     788 LD B,A
C6EE 11 70 C2 789 LD DE,WWWF
C6F1        790 DIRSLP
C6F1 79     791 LD A,C
C6F2 B0     792 CP B
C6F3 D8     793 RET C
C6F4 78     794 LD A,B
C6F5 12     795 LD (DE),A
C6F6 13     796 INC DE
C6F7 AF     797 XOR A
C6F8 12     798 LD (DE),A
C6F9 1B     799 DEC DE
C6FA CD A6 C7 800 CALL FNPRTS
C6FD 04     801 INC B
C6FE 18 F1 802 JR DIRSLP
C700        803
C700        804 ;
C700        805 ; CREV:SCRN
C700        806 ;
C700        807 CRPRTS
C700 DD 5 808 PUSH DE
C701 3A 68 C2 809 LD A,(DU)
C704 5F     810 LD E,A
C705 3A 7C C2 811 LD A,(CB2)
C708 BB     812 CP E ;IF CB<DU
C709 38 14 813 JR C,CRPSK
C70B 57     814 LD A,(YS)
C70C 3A 6C C2 815 LD A,(YS)
C70F 83     816 ADD A,E
C710 BA     817 CP D ;IF DU+YS<CB
C711 38 0C 818 JR C,CRPSK
C713 7A     819 LD A,D
C714 11 70 C2 820 LD DE,WWWF
C717 12     821 LD (DE),A
C718 13     822 INC DE
C719 AF     823 XOR A
C71A 12     824 LD (DE),A
C71B 1B     825 DEC DE
C71C CD A6 C7 826 CALL FNPRTS
C71F        827 CRPSK
C71F 3A 69 C2 828 LD A,(CR)
C722 11 70 C2 829 LD DE,WWWF
C725 12     830 LD (DE),A
C726 13     831 INC DE
C727 3E 08 832 LD A,8
C729 12     833 LD (DE),A ;CREV 1
C72A 1B     834 DEC DE
C72B CD A6 C7 835 CALL FNPRTS
C72E 3A 6D C2 836 LD A,(UU)
C731 57     837 LD D,A
C732 3A 69 C2 838 LD A,(CR)
C735 92     839 SUB D
C736 ED 5B 7A C2 840 LD DE,(CBAD)
C73A 12     841 LD (DE),A
C73B D1     842 POP DE
C73C C9     843 RET
C73D        844
C73D        845 ;
C73D        846 ; LIMIT
C73D        847 ;
C73D        848 LIMITS
C73D C5     849 PUSH BC
C73E D5     850 PUSH DE
C73F 1A     851 LD A,(DE)
C740 FE 80 852 CP $80
C742 D2 78 C7 853 JP NC,UPLMT
C745        854
C745        855 LD C,A
C746 3A 69 C2 856 LD A,(CR)
C749 81     857 ADD A,C
C74A 32 69 C2 858 LD (CR),A
C74D 4F     859 LD C,A
C74E 4D     860 LD A,(DE)
C751 B9     861 CP C ;IF DU+YS<CR
C752 30 03 862 JR NC,LMTSK
C754 32 69 C2 863 LD (CR),A
C757        864 LMTSK
C757 3A 69 C2 865 LD A,(CR)
C75A 4F     866 LD C,A
C75B 3A 68 C2 867 LD A,(DU)
C75E 47     868 LD B,A
C75F 3A 6C C2 869 LD A,(YS)
C762 80     870 ADD A,B
C763 B9     871 CP C ;IF DU+YS<CR
C764 D2 A3 C7 872 JP NC,LMTSK
C767 3A 6C C2 873 LD A,(YS)
C76A 47     874 LD B,A
C76B 3A 69 C2 875 LD A,(CR)
C76E 90     876 SUB B
C76F 32 68 C2 877 LD (DU),A
C772 CD 00 C2 878 CALL DIR
C775 C3 A3 C7 879 JP LMTQ

```

```

C778        880
C778        881 UPLMT
C778 3A 6D C2 882 LD A,(UU)
C778 47     883 LD B,A
C77C 3A 69 C2 884 LD A,(CR)
C77F 4F     885 LD C,A
C780 1A     886 LD A,(DE) ;K
C781 81     887 ADD A,C
C782 FE 80 888 CP $80
C784 30 08 889 JR NC,UPLMTSK1
C786 B8     890 CP B
C787 38 05 891 JR C,UPLMTSK1
C789 32 69 C2 892 LD (CR),A ;CR=CR+K
C78C 18 04 893 JR UPLMTSK2
C78E        894 UPLMTSK1
C78E 78     895 LD A,B ;CR=UU
C78F 32 69 C2 896 LD (CR),A
C792        897 UPLMTSK2
C792 3A 68 C2 898 LD A,(DU)
C795 4F     899 LD C,A
C796 3A 69 C2 900 LD A,(CR)
C799 B9     901 CP C
C79A D2 A3 C7 902 JP NC,LMTQ
C79D 32 68 C2 903 LD (DU),A
C79F CD 00 C2 904 CALL DIR
C7A3        905
C7A3        906 LMTQ
C7A3 D1     907 POP DE
C7A4 C1     908 POP BC
C7A5 C9     909 RET
C7A6        910
C7A6        911 ;
C7A6        912 ; PRT MEM(WORK)
C7A6        913 ;
C7A6        914 FNPRTS
C7A6 CD E1 C8 915 CALL FNMAKE
C7A9 CD C6 C7 916 CALL FNP
C7AC C9     917 RET
C7AD        918
C7AD        919 ;
C7AD        920 ; FN->WORK
C7AD        921 ;
C7AD        922 FNPWKS
C7AD D5     923 PUSH DE
C7AE 11 70 C2 924 LD DE,WWWF
C7B1 3A 69 C2 925 LD A,(CR)
C7B4 12     926 LD (DE),A
C7B5 CD E1 C8 927 CALL FNMAKE
C7B8 3A 69 C2 928 LD A,(CR)
C7BB B7     929 OR A
C7BC 28 06 930 JR Z,FNWRKQ
C7BE 11 54 C2 931 LD DE,WORK+7+13
C7C1 3E 2E 932 LD A," "
C7C3 12     933 LD (DE),A
C7C4        934 FNWRKQ
C7C4 D1     935 POP DE
C7C5 C9     936 RET
C7C6        937
C7C6        938 ;
C7C6        939 FNP
C7C6 C5     940 PUSH BC
C7C7 D5     941 PUSH DE
C7C8 E5     942 PUSH HL
C7C9 3A 68 C2 943 LD A,(DU)
C7CC 4F     944 LD C,A
C7CD 1A     945 LD A,(DE) ;CURSOR POS.
C7CE 91     946 SUB C
C7CF 6F     947 LD L,A
C7D0 3A 75 C2 948 LD A,(USP)
C7D3 85     949 ADD A,L
C7D4 6F     950 LD L,A
C7D5 3A 76 C2 951 LD A,(YY)
C7D8 85     952 ADD A,L
C7D9 6F     953 LD L,A
C7DA 26 00 954 LD H,0 ;YY+N-DU+USP
C7DC 13     955 INC DE
C7DD 1A     956 LD A,(DE)
C7DE 32 71 C2 957 LD (WWF+1),A
C7E1 3A 6E C2 958 LD A,(XX)
C7E4 5F     959 LD E,A
C7E5 16 00 960 LD D,0
C7E7 CD 63 CA 961 CALL XYADR ;BC=ADR
C7EA CD 84 CA 962 CALL TURBO
C7ED DD 21 B2 CA 963 LD IX,WKPT
C7F1 3E 28 964 LD A,40 ;MOJISU
C7F3 DD 77 00 965 LD (IX+0),A
C7F6 CD 10 C8 966 CALL ATR2
C7F9 21 40 C2 967 LD HL,WORK
C7FC        968 ACKLP1
C7FC DD 70 82 969 LD (IX+2),B
C7FE CD 55 C9 970 CALL ACKPT
C780 DD 46 02 971 LD B,(IX+2)
C805 23     972 INC HL
C806 03     973 INC BC
C807 DD 35 00 974 DEC (IX+0)
C80A 20 F0 975 JR NZ,ACKLP1
C80C E1     976 POP HL
C80D D1     977 POP DE
C80E C1     978 POP BC
C80F C9     979 RET
C810        980
C810        981 ATR2
C810 AF     982 XOR A
C811 DD 77 03 983 LD (IX+3),A
C814 DD 77 04 984 LD (IX+4),A
C817 3A 71 C2 985 LD A,(WWF+1)
C81A F6 07 986 OR 7 ;WHITE
C81C DD 77 01 987 LD (IX+1),A
C81F E6 08 988 AND B
C821 C8     989 RET NZ
C822 3A 73 C2 990 LD A,(WWF+3)
C825 FE 04 991 CP 4
C827 3E 06 992 LD A,6 ;YELLOW
C829 CA 51 C8 993 JP Z,ATRN2
C82C 3A 74 C2 994 LD A,(WWF+4)
C82F B7     995 OR A,40
C830 3E 03 996 LD A,3 ;MAZENDA
C832 C2 51 C8 997 JP NZ,ATRN2
C835        998
C835        999 LD A,(WWF+2)
C838 B7     1000 OR A
C839 28 05 1001 JR Z,ATRNKK
C83B 16 04 1002 LD D,4
C83D DD 72 03 1003 LD (IX+3),D
C840        1004 ATRNKK
C840 3A 73 C2 1005 LD A,(WWF+3)
C843 B7     1006 OR A
C844 C8     1007 RET Z
C845 1E 05 1008 LD E,5
C847 FE 03 1009 CP 3
C849 28 02 1010 JR Z,ATRIXS
C84B 1E 04 1011 LD E,4
C84D        1012 ATRIXS
C84D DD 73 84 1013 LD (IX+4),E
C850 C9     1014 RET
C851        1015
C851        1016 ATRND
C851 DD 77 01 1017 LD (IX+1),A
C854 C9     1018 RET
C855        1019

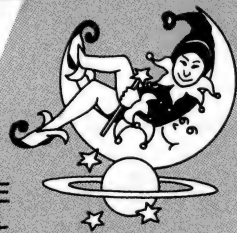
```

▶1年くらい前から仕事でC言語を使っていますが、覚えてたころのソースを見てみると結構笑えるようなことをしていて面白いです。みなさんもたまには昔のソースなど見てみると面白いかもしれませんよ！

宮下 敦(20) 神奈川県

COUPLE

Aoki Michio 青木 実千男



遊び方

CARD.FNCを使ったトランプゲームです。CARD.FNCを拡張したX-BASICからそのまま打ち込んでください。

RUNしてしばらくすると“Push To Start”と出ます。このときカードをシャッフルしているので、少し待ってからマウスのボタン（左）を押す。するとカードが1枚出てきます。次からNextのあたり（このへんおおざっぱ）を押すと、次のカードが出てきて、横に4枚並ぶと次からは1段下がります。Lineが4を越えるとスクロールしていきます。

で、これでいったいなにをしろうかという、COUPLEというひとりて遊ぶトランプゲームなんですよ。

肝心のルールですが、こうして並んだカードが縦、横、斜めに同じ数字なら取っばらっちゃって、空いたところは詰めていくわけです。それで、カードを全部使いきって、全部取っばらっちゃえば“CONGRATULATION”というわけやね（僕は大阪から越えてきてまだ半年なんです。埼玉県加須市在住のX68000ユーザーっていませんか？）。

*

取っばらい方は、（左ボタンで）1枚目クリック。2枚目クリック。ここでも

都合により、先月の予告とは違いますが、ひとり遊びCOUPLEをお届けします。隣接した同じ数のカードを取り除いていく、というルールですべてのカードを取り除いてください。画面いっぱいになるとすーっとスクロールするのが気持ちいい？

う1回クリックするとOKです。キャンセルは右ボタンね。本当の遊び方との違いは、常に最後の4列しか取れないというところ。まあ、遊んでみればわかるでしょう。

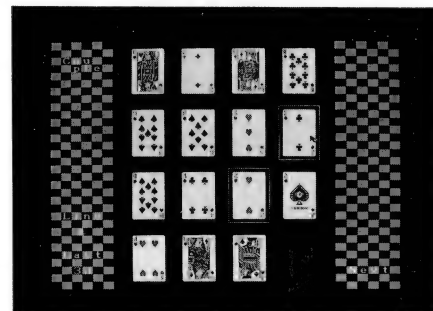
あと、Lineは全部で何行あるか、Lastは残り何枚かを表しています。コツは最初4列くらいばばっと出して、取っては出し、取っては出しを繰り返す、これだと思うんだけど……。

プログラムについて

プログラムの概略は表1を見てもらえばわかるでしょう。プログラムで使用している変数は表2にまとめておきました。それから、サウンドの部分とシャッフルルーチンは1990年3月号の“99”を参考にしています。

表1 内容の説明

90~150	: グローバル変数の宣言
170~230	: これがメインです
250~310	: いろんなものの初期設定
320~390	: 画面の両端にタイトル模様を描きます
400	: そのあとで、ウィンドウエリアを設定し直す
410~450	: card()に順番にカード番号を入れたつ、date()に変数iの全角文字を入れていく
460~510	: 文字を表示
560~590	: 変数その他の初期化
600~660	: カードのシャッフル
730~760	: マウスボタンが押されるのを待つ
770	: Nextを押したとき
780	: カードをクリックしたとき
790	: キャンセルしたとき
860	: カードを出す位置を求める
870	: それが4列目以降、右端のカードならスクロール
1040	: 上下どちらにスクロールするかによって、出てくるカードの位置を求める
1050~1070	: スクロールしたあとに出てくるカードを描く
1080~1100	: スクロール
1160~1190	: 1回目のクリックの処理
1200~1230	: 2回目のクリックの処理
1240~1340	: 3回目のクリックの処理
	カードを取って詰めます
1330	: 4列目以降、左から2枚以内にカードがあったなら逆スクロール
1380~1420	: キャンセルしたときの処理
1460~1480	: カーソル表示
1520~1530	: ウェイト処理
1570~1650	: 2回目のクリックを認めるのかチェック
1580	: そこにカードがあるか
1590	: 数は同じか



ます。

このゲームは夏休みにちょこちょこ作りました。それから、X-BASICはあんまりいじったことがなかったので苦勞したよー。あーあ勉強しなきゃ。いま徹夜明けなので文がしっちゃかめっちゃかだけど許してくださいませ。

1600	: 1回目のクリックと同じカードか
1610~1630	: 縦、横、斜めのいずれかに位置するか
1720	: クリアしたなら……
1730~1830	: ゲームオーバーの処理
1860~1890	: クリアしたときの文字の表示
1900~1930	: 効果音

表2 変数表

グローバル変数

card()	: カードが出てくる順番にカード番号が入っている
scrn()	: 画面に出てきたカードの位置を記憶しておく
pows()	: クリアしたとき、文字拡大の倍率
date()	: 数字の全角文字が入る
ch	: “CONGRATULATION”, クリアしたときに使う
pan	: クリアしたときに使う音
pon	: カードが出たとき、または動いたときに出る音
hy	: homeのY座標の位置
frg	: 何回目のクリックか
hip	: 次にカードが出る位置
dm	: マウス処理のダミー
mx	: マウスのX座標
my	: マウスのY座標
bl	: マウスの左ボタン
br	: マウスの右ボタン
pol	: 1回目をクリックしたカードの位置
kazu1	: そのカードの数
po2	: 2回目をクリックしたカードの位置
kazu2	: そのカードの数
num	: 何番目のカードか


```

10 /* =====
20 /* ===== 至高の CARD.FNC 用ゲーム =====
30 /* =====
40 /* ===== Couple =====
50 /* =====
60 /* ===== 青木実千男 =====
70 /* =====
80 /*
90 dim int card(51),scrn(55),pows(4)=[0,1,6,17,36]
100 dim str data(51)
110 str ch="CONGRATULATION",pan="@68o3v1514c",pon="@59o3v1514c"

120 int hy,frg,hip
130 int dm,mx,my,bl,br
140 int pol,kazu1,po2,kazu2
150 int num
160 /* =====
170 init()
180 while 1
190   cinit()
200   game()
210   gameover()
220 endwhile
230 end
240 /* ----- 初期設定
250 func init()
260   int x,y,c=0,i
270   screen 1,0,1,1
280   console 0,32,0
290   mouse(0):mouse(4):mouse(1)
300   window(0,0,1023,1023)
310   m_alloc(1,200):m_assign(1,1)
320   for y=0 to 63
330     c=(y mod 2)*14
340     for x=0 to 5
350       if c=14 then c=0 else c=14
360       fill(x*16,y*16,x*16+15,y*16+15,c)
370       fill(x*16+415,y*16,x*16+415+15,y*16+15,c)
380     next
390   next
400   window(112,0,398,1023)
410   for i=0 to 51
420     card(i)=i+1
430     if i<10 then { data(i)=" "+chr$(130)+chr$(i+79)+" "
440       } else data(i)=chr$(130)+chr$(val(left$(str$(i),1))+
79)+chr$(130)+chr$(val(right$(str$(i),1))+79)
450   next
460   color 7
470   locate 2, 2:print" C o u";
480   locate 2, 3:print" p l e";
490   locate 2,22:print" L i n e";
500   locate 2,27:print" L a s t";
510   locate 54,29:print" N e x t";
520 endfunc
530 /* ----- カードの初期化
540 func cinit()
550   int a,b,c,i
560   wipe():home(0,0,0)
570   for i=0 to 54:scrn(i)=54:next
580   hip=-1:hy=0:frg=0:num=0
590   score()
600   locate 19,10:print" P u s h   T o   S t a r t";
610   repeat
620     a=rnd():*52:b=rnd():*52
630     c=card(a):card(a)=card(b):card(b)=c
640     msstat(dm,dm,bl,br)
650     until bl=-1
660     locate 19,10:print spc(26);
670   endfunc
680 /* ----- ゲーム
690 func game()
700   for num=0 to 51
710     shori()
720     while 1
730       repeat
740         msstat(dm,dm,bl,br)
750         until bl=-1 or br=-1
760         mspos(mx,my)
770         if mx>416 and my>432 and frg=0 then wait(200):break
780         if mx>111 and mx<400 and bl=-1 then toru()
790         if br=-1 then cancel()
800         wait(700)
810       endwhile
820     next
830   endfunc
840 /* ----- カードを出す
850 func shori()
860   hip=hip+1
870   if hip>15 and (hip mod 4)=0 then scroll(+1)
880   card_set(hip,card(num))
890   score()
900 endfunc
910 /* ----- インフォメーション
920 func score()
930   locate 4,24:print data(hip/4+1);
940   locate 4,29:print data(51-num);
950 endfunc
960 /* ----- カード表示
970 func card_set(po,n)

```

```

980   oto(pon):scrn(po)=n
990   c_put((po mod 4)*72+124,(po/4*128+16) mod 1024,n)
1000 endfunc
1010 /* ----- スクロール
1020 func scroll(d:int)
1030   int i,sy
1040   if d=1 then sy=hip/4+1 else sy=hip/4-4
1050   for i=0 to 3
1060     c_put(i*72+124,(sy*128+16) mod 1024,scrn(i+sy*4))
1070   next
1080   for i=1 to 128
1090     hy=hy+d:home(0,0,hy mod 1024)
1100   next
1110 endfunc
1120 /* ----- カード指定
1130 func toru()
1140   int st,ch,d=1,i
1150   switch frg
1160     case 0:pol=(mx-112)/72+(my/128+hy/128)*4
1170       kazu1=scrn(pol) mod 13
1180       if scrn(pol)<54 then set(pol,11):frg=1
1190       break
1200     case 1:po2=(mx-112)/72+(my/128+hy/128)*4
1210       kazu2=scrn(po2) mod 13
1220       if setcheck()=1 then set(po2,11):frg=2
1230       break
1240     case 2:set(pol,0):set(po2,0)
1250       frg=0
1260       if pol<po2 then st=pol:ch=po2-1 else st=po2:ch=pol-1
1270       for i=st to hip-2
1280         if i=ch then d=2
1290         card_set(i,scrn(i+d))
1300       next
1310       card_set(hip-1,54):card_set(hip,54)
1320       if hip>15 and (hip mod 4)<2 then scroll(-1)
1330       hip=hip-2
1340       score()
1350   endswitch
1360 endfunc
1370 /* ----- キャンセル
1380 func cancel()
1390   switch frg
1400     case 1:set(pol,0):frg=0;break
1410     case 2:set(po2,0):frg=1
1420   endswitch
1430 endfunc
1440 /* ----- カーソル表示
1450 func set(po,c)
1460   int x,y
1470   x=(po mod 4):y=po/4
1480   box(x*72+118,(y*128+8) mod 1024,x*72+176,(y*128+118) mod 1024,c)
1490 endfunc
1500 /* ----- ウェイト
1510 func wait(l)
1520   int i
1530   for i=0 to l:next
1540   endfunc
1550 /* ----- セットのチェック
1560 func setcheck()
1570   int x1,y1,x2,y2
1580   if scrn(po2)<54 then {
1590     if kazu1=kazu2 then {
1600       if not(pol=po2) then {
1610         x1=pol mod 4:y1=pol/4
1620         x2=po2 mod 4:y2=po2/4
1630         if x2>x1-2 and x2<x1+2 and y2>y1-2 and y2<y1+2 then
n return(1)
1640       } }
1650     return(0)
1660   endfunc
1670 /* ----- ゲームオーバー処理
1680 func gameover()
1690   int i,q
1700   str s
1710   wipe():home(0,0,0)
1720   if scrn(0)=54 then s=ch else s=" REGRET"
1730   symbol(144,632,s,2,1,1,15,0)
1740   for i=1 to 384:home(0,0,i):wait(0):next
1750   if scrn(0)=54 then {
1760     for i=0 to 13
1770       oto(pan)
1780       for q=1 to 4:moji(i,q,15):moji(i,q,0):next
1790       for q=1 to 3:moji(i,5-q,15):moji(i,5-q,0):next
1800       moji(i,1,15)
1810     next
1820   }
1830   for i=385 to 896:home(0,0,i):wait(0):next
1840 endfunc
1850 /* ----- ゲームオーバーの文字表示
1860 func moji(x,y,c)
1870   y=pows(y)
1880   symbol(144+16*x,640-y*8,mids(ch,x+1,1),2,y,1,c,0)
1890 endfunc
1900 /* ----- 効果音
1910 func oto(m:str)
1920   m_init():m_trk(1,m):m_play()
1930 endfunc

```

猫とコンピュータ マニアの砦にて

Takazawa Kyoko
高沢 恭子



10月14日に秋葉原で行われた「第7回ホビーマイコンショウ」。キョウコさんも久しぶりに会った通信仲間と、楽しく1日を過ごしたようです。和気あいあいのこの様子を、2回にわたってレポートします。

まだ夏の暑さが見えかくれしている10月はじめの昼さがり。

近くの文具屋さんでケント紙を買いこんで、自転車のカゴにボンと入れたとき、そうだ、プリンタ用紙を買っておくように夫に言われたのだったと思い出した。15インチのほうだったかな、10インチだったかなと考えながら、地下鉄東西線N駅のアーケード街に沿った道にまわると、ペンキ屋さんのしごとに出会った。

改装中の店舗の外壁を塗装しているところで、飛沫をさけるための大きなビニールを広い範囲にめぐらせた中で、脚立に乗った若い人が作業をしている。

よく見ると、壁一面に銃で撃ちつけたような塗料のツブツブがあり、その上からハンドローラーをかけて、表面をたいらに整えている。

ああ、不揃いの基石にプレスをかけたような壁は、こうやってこしらえるのかと、自転車を降りてしばらく見学した。

「塗料を吹き付けてから、どのくらいの時間をあけてローラーをかけるのですか」とたずねると、

「わりあいすぐやってしまうんですよ。塗料が流れちゃいますからね」と下で見ていた年長の職人さんが答えてくれた。油絵の具にそっくりの溶剤の匂いが、すこし不透明な日差しの中に溶け込んだようにあたりに広がり、製作中のアイボリー色の壁も、ツナギ服の2人の若い職人も、ほんわり輝いて見える。なんとも心地よい時間だ。

そのとき、となりのブティックから顔見知りの女主人が出てきて、

「アラ、いつもノンキそうでいいわね」と声をかけた。そうだ、用事が鎖のようにつながっていたんだ。返事がわりのテ

レ笑いをして、ペンキ屋さんにお礼をいうと、予定を変えてスーパーに向かった。プリンタ用紙は重量があるから、また出直すことにしよう。

自由のセンタク

「あの、たとえばですけど……スイグルミなんかも……クリーニングしてもらえるものなんですか」

スーパーの一角にあるクリーニング屋さんには、いつもの店員さんが2人、カウンターに立っていた。「ノンキでいい」と言われたすぐあとなのでこの質問にはちょっとためらいがあったが、参考に聞いておきたいことだったし店も空いていてよい機会だ。

「扱ってますよ。でもけっこういけば安くても1,000円はするんですよ」

値段表をまとめた冊子を持ってきて、スイグルミのページを開いて見せてくれた。いちばん上のランクは3,000円以上になっているが、けっきょく実物を見ないと決められないようだ。

本物の毛皮、ホンニャアが来てから、それまでトオルのペットだったスイグルミたち、古参の順に、フル(古)ニャア、シン(新)ニャアと呼ばれていた彼らがどうなったかという、愛着はじゅうぶんあるものの、長いあいだには転勤などを機に処分されてしまっていた。

ただ1匹、ホンニャアが来る直前に、ある店でみつけた毛足の長い、すこしばかり気どった青い目の猫のスイグルミだけが、新品のまま押し入れにしまわれていた。ペルシャ猫らしい姿なので、トオルが「ペルニャア」と名づけていた。

ホンニャアが一匹前のオトナになったとき、おもしろ半分にこのペルニャアを彼に見せたところ、いきなり怒ったようにとび

つき、スイグルミと抱き合うようなかたちで前足で押さえつけ、後足で猛烈なキックの連打をした。相手に生命がないことは本能的にわかると思うのだが、トレーニングのつもりにしては迫真の興奮ぶり、おそろしいほどだった。それ以後、ペルニャアを決してホンニャアに見せないようにつとめてきた。

それがちかごろ、誰かが捜しものでもしたときにちょっと位置が入れかわったのか、ペルニャアの入ったビニール袋が押し入れの前面にのぞいていた。わが家の押し入れは、通気のために端がいつも10センチほどあいている。私たちはウツカリしていたが、ホンニャアにしてみれば、宿敵との再会だった。

彼はさっそく袋ごとペルニャアをひきずり出して格闘したらしい。発見されたとき、真新しかったペルニャアは、ビニール袋が破られ、玄関のタタキの上で汚れ色になって倒れていた。

「洗濯機で洗っちゃったかどうか」

もうひとりの店員さんが言った。

「私、孫のスイグルミを洗濯機で洗いましたよ。そのあと3日間、外に干しっぱなしにして乾かしましたけどね」

スゴイ！ なんていう迷いのない力強いやりかた。汚れを落とすならそれがいちばんだ。私はといえば、あのフワフワの風合いを失うまいと、ぬるま湯、毛糸洗剤、手洗い、最短時間、陰干しという臆病な条件で、いまでも何回も挑戦しながら、いつも失望をくりかえしてきたのに。だから、こんどのペルニャアは、汚れ落としより風合いを優先して、ドライクリーニングを考えたのだ。

「それで、仕上がりはどうでしたか？」

「真っ白になりましたよお」

お孫さんがいるにしてはまだとても若いその人は、満足そうに言った。洗濯機の中で目をまわし、3昼夜戸外にさらされて、疲れてカラカラになったヌイグルミが目に見え浮かんできたが、洗濯というのは仕上がりにかける期待によって、こんなに奔放なやりかたがセンタク（選択）できるものなのだと気がついた。

さて、買い物をすませたら、ホビーショウのポスターをつくらなくては。

隠し砦の善人たち

雨の宣告を受けていた10月14日は、朝からだんだん空が明るくなり、日中は汗ばむほどの日となった。

「第7回ホビーマイコンショウ」は、いつものように、秋葉原ラジオ会館8階大ホールで開催された。創立11周年の「きまぐれコンピュータクラブ」と、同じく10周年の「FORESIGHT」（フォーサイト）、それに開局5年目を迎えた「FBI-NET」の共催である。

プロ、アマをまじえたメンバーが、ハード、ソフトのオリジナル作品を披露しながら、親睦と情報交換の1日をするのが目的で、誰でも無料で入場できる。

ただでさえ行事の多い季節に、今年は科学技術館で毎年開かれる「全国草の根BBS大会」が1週間後と、日程が接近してしまい、スタッフも共通していることから、みんなたいへん忙しい思いをした。

いつも「踊る人形」などの人気作が話題のコバヤシ先生は、予定の「文福茶釜」が故障で、「デジタル気圧計」を出品。金沢の高校の数学の先生、ワカマツさんは、継続テーマである「 π の計算」を、前回の15万桁から、100万桁まで算出に成功。FM TOWN SによるCGのデモとあわせて出品された。

イシイさんは、中古基板を組み合わせた自作のゲーム。群馬のタムラさんは、「音声認識の実験」。画面に向かって口笛を吹くと、マイクから入力した音をA/D変換してZ80ボードで処理。PC-6001で同じ音程を再現する。

広島県の理科の先生イマオカさんは、教材としての目的で開発した、MSX2による「圧力測定システム」。測定の結果をグラフ表示もする。ソフト制作が専門のナカニシさんも、CAIの例として、「化学の尺度モル」

の概念をわかりやすく説明するグラフィックを出品、デモをしてみせた。

実業家のオザキさんは、手作りのガイガーミュラー管による「宇宙線センサー」、広い大気中に飛び交う、目に見えない「宇宙線」を感じて音で知らせる。

夫はショウ開催の段取りに時間を取られていたが、人間が

近づくると24の小窓が点滅するイルミネーションのセンサーを、装飾をかねて製作した。

FBI-NETの2大名物は、10回線の端末を使った「パソ通シミュレーション」と、「天麩羅★三杯酢」ことヤマザキさんの「電脳駄菓子屋」さん。

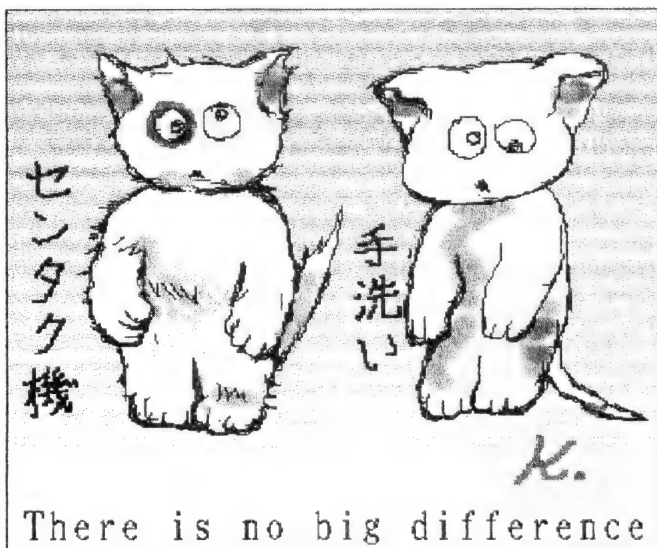
展示のほかに、PA装置完備のトークセッションプログラムとして、BBSの生みの親でもあるヨコタ先生（横田秀次郎氏）の「OS/2 LANマネージャとNetwareの戦い」と題する講演、カワムラ先生（川村清氏）の通信端末プログラムの新モデル「BIG-Term」の発表、ナカニシさんの展示作品「化学の尺度モル」についてのスピーチが行われた。

通信ファミリー

「キョウコさん、エレベータの中のポスターですけれどね、猫がもっているGUNはMGCのM75でしょ」

FBIの「ちやがま」ことナカムラテツヤ君に、また名答されてしまった。ラジオ会館の2基あるエレベータの中に、ホビーショウのポスターとして、1対のコミックを描いた。1枚はICや電子パーツを積んだ要塞から、猫がGUNをかまえているもの。もう1枚は、弾丸を受けて倒れているもので、「マニアの最後の砦」なんてコメントも入れた。

この連載でいつか描いたGUNも、ちやがま君に「あれはPPKですね」とあっさり当てられて驚いたが、くわしく描きこんだ絵ならともかく、簡単に印象だけで描いたものなのに、いかに熟知しているかとい



うことだろう。

FBIのGUNボード「SWAT」とシャレでやっている「きょうこファンクラブ」の両方のシグオペでもあるちやがま君は、この春から業界大手Q社の社員である。

「みゆ」君、「むーみむ」君、「CLOVIS」君。イベントというと、大きな機動力とタレント性を発揮してくれるFBIの主力メンバーの多くが、この春から社会人となった。

画面の文字を通じて初めて交信する感動をいっしょに体験した仲間たち。半角のカタカナから、全角漢字へ、そしてANSIのカラーグラフィックまで。通信が成長していく歳月と併せて、5年間の貴重な時代を共にすごしたことになる。

そういえば、小学生だった「GETIKO BA」ちゃんは今もう高校生、中学生で私にANSIを手ほどきしてくれた「ながみね講師」も来春は大学受験、新聞社のワープロオペレータ「ちやこ」さんは、結婚してママになり、プロ中のプロであるSSKさんも、この間に大手のコンピュータ会社から独立した。

シスオペ「nin隊長」（中村守利氏）の尽力とメンバーの力でネットは熟成し、寂しかった女性ボード「ばななくらぶ」も、一挙に若いメンバーが増えた。

きょうのショウの進行の途中で、その女性ネットワークカーたちを取材させてほしいと、某女性週刊誌から申し込みを受けた。

FBIのメンバー、編集マンの「銀猫」さんを通じての依頼で、約束の3時に記者氏がカメラマンをとまってあらわれた。

（以下次号へ）

BACK ISSUES

バックナンバー案内

ここには1989年12月号から1990年11月号までをご紹介します。現在1989年10～12、1990年2、4、6、8～11月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については、186ページを参照してください。

1989

1990



12月号

特集 Cプログラミングへの招待

付録 C言語簡易リファレンス

連載 ショートプロバート/280's Bar

●X68000マシン語/X-BASIC/D6GA・CGA

●Oh! X2周年特別企画「素粒子の音が聞こえる」

●X1/turbo用アクションゲームACTIVE UNIT

LIVE in '89 天空の城ラピュタ/ギャラクシーフォース

THE SOFTOUCH 38万キロの虚空/た〜みのる2

全機種共通システム Slang用リダイレクションライブラリ



1月号 (品切れ)

特集1 オペレーティングスタイルの研究

特集2 Cプログラミング応用編

連載 ショートプロバート/280's Bar

●X68000マシン語/C調言語講座/D6GA・CGA

●X1/turbo用シミュレーションゲームSuper Battle

LIVE in '90 さよならを過ぎて/Rydeen

THE SOFTOUCH レナム/メタルサイト

全機種共通システム WORM KUN/再掲載SLANG

特別付録 X68000 THE SOFTWARE CATALOGUE



2月号

特集 画像圧縮へのアプローチ

連載 ショートプロバート/280's Bar/D6GA・CGA

●X68000マシン語/C調言語講座/X-BASIC調理実習

●X68000用ゲームプログラムGon Gon

●MZ-700用紙芝居Eylarh

LIVE in '90 オーダイン/魔女の宅急便

THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリス

マジックパレット/Mu-1/CYBERNOTE PRO-68K

全機種共通システム 超小型コンパイラTTC++



3月号 (品切れ)

特集 MUSICアドベンチャー

X68000用MIDIドライバ&音源エディタ

なんでも鳴らせるOPMD.X/MMLを楽譜データに

連載 ショートプロバート/280's Bar/D6GA・CGA

●C調言語講座/X-BASIC調理実習

●X1/turboシミュレーションCRISIS in Tokyo

LIVE in '90 パワードリフト/スキーム/となりのトロ

THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター

全機種共通システム 超多機能アセンブラOHM-Z80



4月号

特集 ゲームシステム文学誌

1989年度GAME OF THE YEAR発表

連載 ショートプロバート/280's Bar/D6GA・CGA

●X-BASIC調理実習/C調言語講座/X68000マシン語

●X1・MZ-2000/2500用RPG The Cave of Dalk

●うわさの68040、ついに登場

LIVE in '90 パーニングフォース(OPMD対応)

THE SOFTOUCH The Fille Professor/HOST PRO-68K

全機種共通システム ファジコンコンピュータシミュレータ-MY



5月号 (品切れ)

特集 BASICプログラミング

第5回 言わせてくれなくちゃだわ

連載 ショートプロバート/280's Bar

●X-BASIC調理実習/X68000マシン語プログラミング

●新機種X68000SUPER-HD/EXPERT II/PRO II

●ラジコンスティックの製作

LIVE in '90 TURBO OUTRUN

THE SOFTOUCH 天下統一/ポピュラス/Hyperword

全機種共通システム インタプリタ言語STACK



6月号

特集 創刊8周年記念PRO-68K(付録5"2HD)

Oh! Xアンケート結果大分析大会

連載 ショートプロバート/280's Bar/PurePASCAL

●X-BASIC調理実習/X68000マシン語プログラミング

●X1 turbo用コマンドシェルシミュレータ

●ハードウェア工作入門

LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木

THE SOFTOUCH 三国志II/FAR SIDE MOON/グラナダ

全機種共通システム X68000用S-OS"SWORD"他



7月号 (品切れ)

特集 マシン語への第一歩

X68000SUPER-HD試用レポート

連載 ショートプロバート/280's Bar/D6GA・CGA

●X-BASIC調理実習/PurePASCAL

●INTEGRAL X1——ノーマルX1への対応

●ハードウェア工作入門

LIVE in '90 夢幻戦士ヴァリスII/トッカータとフーガニ短調

THE SOFTOUCH サーク/あーくしゅ/ダウンタウン熱血物語

全機種共通システム リロケータブルアセンブラWZD



8月号

特集 ADVANCED 2D GRAPHICS

100号記念特別モニタプレゼント

連載 ショートプロバート/280's Bar/INTEGRAL X1

●X-BASIC調理実習/X68000マシン語プログラミング

PurePASCAL/ハードウェア工作入門

●X68000用画像回転プログラム XROT0.X

LIVE in '90 OMENS OF LOVE/ENDLESS RAIN/ダートフォックス

THE SOFTOUCH 大航海時代/ウルティマV/プロミストランド

全機種共通システム リンカWLK



9月号

特集1 日本語を処理するための序章

特集2 ADVANCED 2D GRAPHICS

連載 ショートプロバート/280's Bar/D6GA・CGA

●X-BASIC調理実習/マシン語プログラミング

PurePASCAL/ハードウェア工作入門

●清水和人流プログラミング道場

LIVE in '90 風の谷のナウシカ/ラジオ体操第一

THE SOFTOUCH T&T/D-Again/シミシティー/ギャラガ'88ほか

全機種共通システム BILLIARDS



10月号

特集 電子音楽入門

連載 ショートプロバート/280's Bar/D6GA・CGA

●マシン語プログラミング/ハードウェア工作入門

清水和人流プログラミング道場

●荻窪圭の大人ののためのX68000

●中森章のようこそここへC言語

LIVE in '90 Rise And Fall/PARADOX/キュービー3分クッキング

THE SOFTOUCH ワールドコート/ルーンワース/闇の血族/提督の決断

全機種共通システム ライブラリアンWLB



11月号

特集 理科系のGAME REVIEW

連載 Z80's Bar/D6GA・CGA/カードゲーム

●マシン語プログラミング/ハードウェア工作入門

PurePASCAL/X-BASIC調理実習

ようこそここへC言語/INTEGRAL X1

荻窪圭の大人ののためのX68000

LIVE in '90 ピラミッドソーサリアン/ザ・スキーム

THE SOFTOUCH SPECIAL ラゲーン/幻獣鬼/サイバリアン/GUNSHIP他

全機種共通システム スクリーンエディタEDC-T



郵便はがき

料金受取人払

高輪局承認

1459

差出有効期間
平成4年7月
15日まで

1 0 8 - 0 0

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル
ソフトバンク株式会社

Oh!  編集部行

□□□-□□

電話

住所

氏名

年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特集について

いちばん良かった記事

興味のなかった記事

これから載せてほしい記事内容

本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

あなたのパソコンの拡張スロットには何がささっていますか？

あなたの愛機は(所有機種に○印をつけてください) ない
 X1(マニアタイプ,C,D,F,G,twin) X1turbo(model 10,20,30,40,II,III,Z,ZII,ZIII)
 MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861)
 X68000(初代,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,HD) その他
 FD(基) TAPE QD HD(MB) プリンタ()

年齢 歳 パソコン歴 年 男・女 プレゼントNo.

振替用紙

点線からきれいに切り取ってご使用ながいます。

通常払込料金
加入者負担

払込票

口座番号	東京1	29307	ソフトバンク株式会社	
加入者氏名				
住所				
払込住所氏名				
備考				
受付局日付印				

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

通常払込料金
加入者負担

払込通知票

口座番号	東京1	29307	ソフトバンク株式会社	
郵便番号				
払込住所氏名				
受付局日付印				

この払込通知票は、機械で使用しますので、下部の線を折らないよう特に御注意ください。また、本票を折り曲げたりしないでください。（郵政省）

各欄の印欄は、払込人において記載してください。

定期購読のお申し込みを頂うありがとうございます。
 させていただきます。この申込書の弊社到着後は次の通りです。これを過ぎますと次号からの発送となりますので、ご了承ください。尚、郵便振替は郵便局で払い込まれてから当社に到着するのに、2週間位かかります。

Oh! PC 1日発売 前月 20日
 15日発売 当月 5日

月刊情報処理試験
 BEEP メガドライブ 前月 末日

Oh! X
 Oh! FM
 THE COMPUTER 当月 10日
 C MAGAZINE
 パソコンマガジン

(御注意)
 当月発売の号より前に遡ってからの申し込みは出来ません。

お申込みの際は、お近くの書店で申し込み下さい。

この欄は、加入者毎の通信に利用してください。切り取りないうて郵便局にお出ください。

送り先		性別	年齢	職業
フリガナ		男・女		
フリガナ				
〒				
住所				
支店				
郵便局				

申込書	期購読	新規申し込み	継続申し込み	年 間
THE COMPUTER	定期購読	新規申し込み	継続申し込み TC NO.	年間 7,200円
Oh! PC	定期購読	新規申し込み	継続申し込み PC NO.	年間 11,440円 6~11月 5,720円
Oh! X	定期購読	新規申し込み	継続申し込み X NO.	年間 6,720円
Oh! FM	定期購読	新規申し込み	継続申し込み FM NO.	年間 6,720円
月刊情報処理試験	定期購読	新規申し込み	継続申し込み J.S. NO.	年間 8,160円 6~11月 4,080円
C MAGAZINE	定期購読	新規申し込み	継続申し込み CM NO.	年間 11,760円
パソコンマガジン	定期購読	新規申し込み	継続申し込み PM NO.	年間 6,960円
BEEP メガドライブ	定期購読	新規申し込み		年間 5,760円

この申し込み欄は、機械で使いますので、下部の欄を過ぎないように折に注意してください。また、本票を折り曲げたりしないでください。(郵 政 省)

[第7話]

スキーは豪華に？

TAKAHARA HIDEKI 高原 秀己

秋も残りわずか。

いよいよスキーシーズンの到来、つてとこである。ぼくもここ数年、1シーズン2、3回はスキーに出かけている。しかし、なかなか上達しない。スコア110のゴルフのほうがまだサマになっていると思うほどのだから始末におえない。わずか10分で頂上から高速滑走して一気にふもとまで着いてしまうという芸当など、間違ってもできないのである。

しかも悪いことに、やっていて、さして楽しいとは感じられないのもどうにも困ったところである。決してつまらないと思いつながらやるのではないにせよ、

「これほど楽しいスポーツはないぞ。スキーこそレジャーの極み」

という友人の感覚には、ほど遠い。

ではなぜ、スキーに行くんだ？ ということになるのだが、これはもう、世の中全体が冬になるとスキー、スキーと大騒ぎしていて、ぼくもしなくてはいけないような妄想に囚われているからにはほかならない。

「主体性がない、情けない」なんて声も聞こえてきそうだが、冬の週末に新宿なんかに行ってみると、もう大変。スキーに行く人たちが街はあふれかえっていて、西口方面は駅前から新宿中央公園の南側まで、スキーバスが無限に並んでいる。丸の内なんかと同様だし、果ては大手町や虎ノ門なんかのビジネス街でもスキー板を抱えた、場違いな若い人がウヨウヨしている。

こうなると、ひとりだけ行かないと、なにか社会に参加していないのではないかと、いけな生き方をしているのではないかと、思ってしまったたりもする。

まあここまでいうのはオーバーで、スキーのどこが面白いのかを探っているところ、といえいいのだろうか。

一般的に最近のスキーの特徴といえば、よりぜいたくに、よりファッショナブルに、

という点につきるだろう。行き先は北海道、さらにはカナダ、ヨーロッパ。ウェアもパステルカラーあり、蛍光色あり、今年はダウンジャケット感覚の渋めのものありと、年々にぎやかになってきている。

苗場だ、白馬だ、妙高だ、という東京から近いところは、混んでいることもあって、あまり行っても自慢にならない。自慢にならないだけならいいが、まともに滑れないとあっては問題なのだ。

苗場なんかはひどいもので、昨シーズンは280万もの人が集まってしまったとか。1日2万人の割合だから、すさまじい。実際に行ってみると、ゲレンデは人の洪水、状態がちょっと悪い場所になればコブごとに渋滞、スピードを出して滑ってきた人がコブにとりついていて人とぶつかって大騒ぎ、といった状態。とにかく滑る場所を探すので、もう精一杯なのだ。

これは苗場だけのことではない。万座、白馬などもシーズン100万人級だし、蔵王なんかだとリフトで数十分並ぶ、なんてことも珍しくない。

その点、北海道はさすがに違う。ゲレンデ自体が広いこともあるが、平日なんかだと、ゲレンデのうち視界に入る部分にはわずか5人、という余裕。上手な人でも下手な人でも、近場よりははるかにいいコンディションで滑れるというわけだ。

「そんなのみんなわかっているんだから、ドバツと人が押し寄せて同じことになるはずだろう」

という人もいるのだが、さにあらず。

なにせ北海道とあって、空路が中心。飛ぶ飛行機の数を決まっているのだから、いわゆる“衝動スキー”組は行けない。

しかも札幌ルートにせよ旭川ルートにせよ、足以上に宿に限度があって、パックスアーなんか自然に制限がかかってしまう状態になっているという。

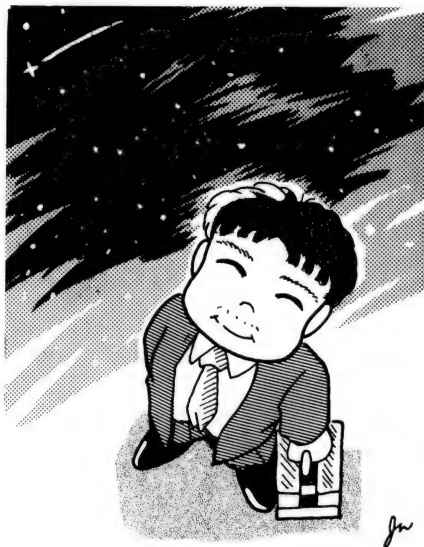
北海道にいるということもあって気分だけでも全然違うのだが、環境、混雑状況までいいとあっては、人気は出るはずである。

この北海道スキーをもう一步、ゴージャスにしたのが流行しはじめている海外スキーだ。なにしろディスカウント合戦の旅行代理店業界とあって、6日間で12万円ちょっと、というカナダスキーツアーまでお目見えしている。

まあこれはシーズン外れゆえの特別価格ではあるのだが、それにしても15万円前後でカナダに行ってスキーができるのだから、恐ろしい話である。

近場で4万円なら6万円出して北海道に、6万円出して北海道なら15万円出してカナダに、というぜいたく感覚、たいしたことがないように思えるのは自然なのだろう。儉約家の人でも近場2回を北海道1回に、ということなら問題はないのだろうし。

と、ここまで書いて改めて感じるのだが、スキーとはいっても、メンバーとか夜の部の活動内容(?)によっても、楽しさたるや全然違うことはいうまでもない。実際、「スキー合宿」と「スキーパーティ」とは天と地ほど違うのだ。



Oh! INDEX'90

特集

オペレーティングスタイルの研究

OSとオペレーティングスタイル	1, 48
For Beautiful Human58k Life	
基本コマンド攻略法	1, 50
コマンドひとつで全自動	
基礎から学ぶパッチファイル	1, 57
コマンドに歴史あり	
マジカル・ヒストリー・ツアー	1, 61
自分だけのおいしいコマンドライン	
エイリアス主義のすすめ	1, 65
コマンド操作からシステムコールの使い方まで	
OS-9 プログラミング教室所	1, 67

Cプログラミング応用編

数式記述から3Dタートルまで	
再帰大作戦	1, 118
8ビットでもC言語を	
αCで書く正規表現	1, 126
ここまでできる	
GCCに見る最適化	1, 131

画像圧縮へのアプローチ

自然画像の圧縮に挑む (1)	
YC分離とADPCM	2, 50
自然画像の圧縮に挑む (2)	
ウォルシュ変換とアダマール変換を使う	2, 56
アルゴリズム解説	
これが噂のPIC R	2, 75

MUSICアドベンチャー

データの互換性を探る	
ミュージックメディアの可能性	3, 34
X68000用MIDIドライバ	
MIDIDRV, SYS	3, 38
Oh!X'90年型ミュージックドライバ	
なんでも鳴らせるOPMD, X	3, 43
Roland MT-32, CM-32L, CM-64, D-10/20用	
LA音源用音色エディタ	3, 60
KORG M1シリーズ用	
AI音源用音色エディタ	3, 71
X1 MIDIボード用	
MIDIデータローダ&セーバ	3, 74
OPMファイルto MUSIC PRO-68K	
MMLを楽譜データに	3, 78

ゲームシステム文学史

ワンダラーズ・フロム・イース	4, 82
ファーストクィーン	4, 85
アークスII	4, 88
サンダーブレード	4, 90
バブルボブル	4, 92
Zero 第4のユニット4	4, 94
人気ゲームのシステムを探る	4, 97
アーケードに見るゲームシステム	4, 102
ゲームセンターとパソコンゲームのふかい関係	4, 102
ダンジョンマスター	
泉は一度だけ願いをかなえる	4, 106
スーパーハンガオンとレースゲームの未来	4, 110
ゲームミュージックを斬る	4, 113
ゲームソフト進化論	4, 116
次世代を担うゲームのテーマ	4, 118

BASICプログラミング

入門者のための	
X-BASICの心得	5, 74
潜入! バグ対策24時間	
X-BASICはいま……!	5, 76
X-BASICでゲームを作る	

豪華版 SCRAMBLE	5, 80
CARD, FNCを活用する	
カードゲームを作ろう	5, 83
Z80's Bar番外編	
通信によるファイル転送	5, 87
アルゴリズムを考える	
拡大縮小処理の基本	5, 92
BASICで作るXCオプティマイザ	
プロトタイピングのすすめ	5, 97

創刊8周年記念PRO-68K

便利なツールあれこれ	6, 50
リアルタイムパズルゲーム	
Yet Another column	6, 51
音楽演奏関係	
OPMD & MUSICDRV	6, 52
X68000対応GNUコンパイラ	
GCC Ver. 1.36.01	6, 54
ウィルス検出プログラム	
DOCTOR, R	6, 56
インテリジェントソースジェネレータ	
DIS, X	6, 58
対話型CGAシステム	
ANGEL	6, 62
SX-WINDOW & キャンバス, X用	
グラフィックデータコンバータ	6, 68
アフターケア	
創刊8周年記念PRO-68Kのその後	7, 130

マシン語への第一歩

ぜんまいちゃん再び	7, 42
マシン語ってなに?	7, 47
MC68000の動作を探る	7, 52
基本装備とおまじない	7, 58
DOSコール&IOCSコールを使う	7, 63
マルチタスクへの挑戦	7, 67
S-OS & REDAを忘れていませんか	7, 74

ADVANCED 2D GRAPHICS

X68000用グラフィックツール紹介	
あなたにあったグラフィックツール	8, 44
ギザギザのないグラフィック関数	
アンチエイリアスとは?	8, 50
X-BASICによる画像処理	
後処理によるジャギーの除去	8, 68
色数の補間と量子化	
グラフィックデータを変換する	8, 72
4096色→8色変換	
Zの画像をX1で	8, 77

日本語を処理するための序章

ワープロを使う前に	
日本語を書くための7つの方法	9, 46
X68000の日本語環境を見る	
我慢せずに使うWP, X	9, 51
雷語! 号はどうなるのか?	
ホメオスタシスへの道	9, 56
ASK68K用辞書整備メンテナンスツール<前編>	
辞書整備基本編	9, 58

ADVANCED 2D GRAPHICS総論

デジタルペインティングへの道	9, 113
----------------	--------

電子音楽術入門

PIXYとX68000とMML	
FM音源のある部屋	10, 82
システム上でBGMを	
OPMファイルで遊ぼう	10, 84
初心者から中級者に贈る	
音楽プログラムは怖くない	10, 86
FM音源用MML記述テクニック	
Yこそすべて	10, 90

多彩な表現をマクロにする外部関数

ZMUSIC, FNC	10, 97
付録 簡易OPMファイルジェネレータ	
OMUSIC, FNC	10, 113

理科系のGAME REVIEW

コンピュータから見たゲームの世界

吾輩はパソコンである〜ゲームに魅せられた主の話	11, 98
ゲームと認知 プレイヤーの時空間	
神よ、私の時間が揺らいでも私はまだ生きている	11, 102
ゲーム空間のメタ理論	11, 106
Wizardryに見るゲームの楽しさ	11, 108
Column ゲームはアプリケーションだったんだ!	11, 108
ゲームデザインとその表現	
フライトシミュレーションのあるべき姿を探る	11, 110
敵キャラクターをもっと大切に	11, 113
ゲーム作成システムを考える	
あなたがゲームを作れない理由	11, 115

XCのための傾向と対策

Cコンパイラのアウトライン	
XC ver. 2.0ガイドマップ	12, 106
コラム Cコンパイラが取り扱うファイル	12, 111
XCを支援するおいしいツール	
ソースコードデバッガを使ってみよう	12, 112
コラム 使って便利な外部コマンド	12, 115
貴方のプログラミングを支援する	
線の下のプロセッサ	12, 116
基礎知識からプログラミングへ	
Cライブラリ活用の手引き	12, 121
多数のソースファイルを管理する	
XCにMAKEが付いてきた	12, 127
MIDI制御が加わった	
新しい音楽ドライバOPMDRV 2, X	12, 129
BASICコンパイラ派に贈る便利ソフト	
XBASToC CHECKER	12, 131

特別企画

第5回日本列島縦断マラソン

カラーイラスト大集合	
Oh!X readers'ぎやらい	5, 28
Micro Communication	
言わせてくれなくちゃだワ	5, 125
どんな悩みもスッキリ解消	
ざ・質問箱SPECIAL	5, 138

創刊8周年記念

第1回Oh!Xアンケート結果大分析大会	6, 102
創刊8周年記念愛読者特大プレゼント	6, 109

通巻100号記念

表紙ぎやらい	8, 23
対戦ポビュラス 祝一平VS西川善司	8, 97
愛読者特大モニタプレゼント	8, 100

Oh!X 3周年記念

愛読者プレゼント	12, 40
アナログジョイスティックの製作	12, 86

THE SOFTOUCH

THE SOFTOUCH SPECIAL

1989年度GAME OF THE YEARノミネート作品発表	1, 30
1989年度GAME OF THE YEAR発表	4, 28
またまた勝手にGAME OF THE YEAR	4, 36
話題のゲームソフト大公開	11, 38
新作ゲーム特大レビュー	12, 46

新作ソフト情報

ウルティマIII/A-JAX/モトス/信長の野望・戦国群雄伝/Misty2/ダンジョンマスター/Musicstudio Mu-1	1, 34
セレクトッドソーサリアン1/アルガーナ/銀河英雄伝説/上海II/水樹伝/ディオス/南海の死闘/The File Professor	2, 24
セレクトッドソーサリアン2/アークスII/あ〜くしゅ/神戸恋愛物語/バブルボブル/ワンダラーズ・フロム・イース/Misty3/ずるかまし/HOST PRO-68K	3, 18
キューブランナー/グランディアフロム/サンダーブレード/ダウタウン熱血物語/ポビュラス/プロデューサー/スライマ/タッグ・オブ・ウォー/銀河英雄	

雄伝説 パワーアップ&シナリオ集/グラナダ	4, 44
Misty 4/ギルガメッシュ・ソーサリアン/闇の血族/クオース/ジェミニウイング/サーク/リフライム/ブレード・オブ・ザ・グレート・エレメンツ/サイクロン Express α	5, 44
ファーサイドムーン/びんびん麻雀ビーチエンジェル/維新の嵐/ルーンワース〜黒衣の貴公子〜/バズニック/アソコの幸福/天九牌・桃源の宴/チェイス H.Q./ワールドコート	6, 42
天下統一/天九牌 Special 桃源の宴Part 2 女子高生編/風神魔伝II/PLANET/フェブリー/大航海時代/ウルティマV/ユニオン/REINFORCER/G-TOOL/実践ピリヤード/プロミストランド/C-TRACE ver. 3	7, 38
ギャラガ'88/遊撃王II/Thrice/サイバリオン/ラグーン/幻獣鬼/イメージファイト/バルーサの復讐	8, 38
トンネルズ&トロールズ/提督の決断/D-Again/エメラルド伝説/雀豪2/PIPYAN/FSSシナリオvol.1/ペルセウスの冒険/Vessel/シムシティー/アクシス/びくせる君	9, 34
三国志II/セレクトッドソーサリアン5/ヴァークチュア・ナイト/FLY/GUNSHIP/Misty vol.5/機甲師団/ブル・オブ・レイディアンス/熱血高校ドッジボール部サッカー編	10, 36
ランペルール/PINBALL・PINBALL/ハイドライド3SV/G-TOOL/XBASTOC CHECKER PRO-68K/ニューラルギア/銀河英雄伝説II/ダンジョンマスター・カオスの逆襲/CANVAS PRO-68K	11, 52
NAIOUS/LUCY・SHOT/3D倶楽部/ダイナマイトデューク/パロディウスだ! /選べるオーガスタ/シュヴァルツシルト/リングマスターII 永遠なる想い/生中継68/ブルトン・レイ/ラブラの魔/栄冠は君に/アトミックロボキッド/ワールドスタジアム/C-TRACE68+	12, 44
GAME REVIEW	
ヒーロー・オブ・ランス/ウルティマIII/シャッフルバック・カフェ/フラッピー2/V'BALL/モトス/倉庫番パーフェクト	1, 36
アルガーナ/信長の野望・戦国群雄伝/アルビオン/バトルチェス/ナイトアームズ/ダンジョマスター/スーパーハンガオン	2, 26
水滸伝/銀河英雄伝説/ディオス	3, 20
やじうまベナンレース/上海II/レナム/夢幻戦士ヴァリスII/ガンマ・ブラネット	4, 44
AFTER REVIEW	
バブルボブル/サンダーブレード/ファーストクイーン/アルガーナ	5, 44
ダンジョンマスター/ポピュラス/ワンダラーズ・フロム・イース	6, 42
グラナダ/FAR SIDE MOON/サーク/三国志II	7, 38
天下統一/ダウタウン熱血物語/あ〜くしゅ/Yet Another Column	8, 38
大航海時代/プロミストランド/ウルティマV/SX-WINDOW	9, 34
ギャラガ'88/トンネルズ&トロールズ/クオース/D-Again	10, 36
GAME REVIEW (SPECIAL REVIEW)	
レナム	1, 40
メタルサイト	1, 42
A-JAX	2, 30
フラッピー2	2, 32
夢幻戦士ヴァリスII	2, 34
ウルティマIII	2, 36
ナイトアームズ	3, 22
新 [ZAN] 〜陽炎の時代〜	3, 24
ダンジョンマスター	3, 26
ゲームシステム文学誌	(→特集4)
天下統一	5, 32
ワンダラーズ・フロム・イース	5, 34
グラナダ	5, 36
ポピュラス	5, 38
ダンジョンマスター	5, 41
三国志II	6, 34
FAR SIDE MOON 地球防衛軍II	6, 36
グラナダ	6, 38
ワンダラーズ・フロム・イース	6, 40

サーク	7, 32
あ〜くしゅ	7, 34
ダウタウン熱血物語	7, 36
大航海時代	8, 32
ウルティマV	8, 34
プロミストランド	8, 36
トンネルズ&トロールズ	9, 24
D-Again	9, 26
シムシティー	9, 28
ギャラガ'88	9, 30
クオース	9, 31
プロテニス ワールドコート	10, 28
ルーンワース〜黒衣の貴公子〜	10, 30
闇の血族	10, 32
提督の決断	10, 34
ラグーン	11, 38
幻獣鬼	11, 40
サイバリオン	11, 42
GUNSHIP	11, 44
ランペルール	11, 46
ティグナスの冒険	11, 48
雀豪2/びんびん麻雀ビーチエンジェル	11, 50
ピンボール・ピンボール	11, 52
パズルゲーム再考(上海II/ユニオン/リフライム/パイピヤン/キューブランナー/プロディア/バズニック/Yet Another Column/スライス/クオース/スライヤン)	11, 54
イメージファイト	12, 46
ジェミニウイング	12, 48
NAIOUS	12, 50
FZ 戦記 アクシス	12, 52
機甲師団	12, 54
ニューラルギア	12, 56
闇の血族完結編	12, 57
熱血高校ドッジボール部サッカー編	12, 58
エアー・コンバット(遊撃王II)	12, 60
バルーサの復讐	12, 62
SPECIAL REVIEW	
マジックバレット	2, 38
Musicstudio Mu-1	2, 40
C-FORM	2, 42
CYBERNOTE PRO-68K	2, 44
The File Professor	4, 40
HOST PRO-68K	4, 42
Hyperword	5, 46
サイクロンExpress α	6, 44
Communication PRO-68K ver. 2, 0	9, 32
C-TRACE68 ver. 3, 0/C-TRACE68 TP	10, 38
最新グラフィックツール試用レポート	
CANVAS PRO-68K/G-TOOL	11, 31

シリーズ全機種共通システム

THE SENTINEL	1, 139
SLANG用ゲームWORM KUN	1, 140
再掲載SLANGコンパイラ	1, 145
THE SENTINEL	2, 81
超小型コンパイラTTC++	2, 82
THE SENTINEL	3, 129
超多機能アセンブラOHM-Z80	3, 130
THE SENTINEL	4, 149
ファジィコンピュータシミュレーターI-MY	4, 150
THE SENTINEL	5, 105
インタプリタ言語STACK	5, 106
THE SENTINEL	6, 113
リロケータブルフォーマットの取り決め	6, 114
STACK用ゲームSQUASH!	6, 118
X68000用S-OS "SWORD"	6, 122
特別付録 PC-286/9801用S-OS "SWORD"	6, 128
THE SENTINEL	7, 77
リロケータブルアセンブラWZD	7, 78
THE SENTINEL	8, 145
リンカWLK	8, 146
THE SENTINEL	9, 145
BILLIARDS	9, 146

THE SENTINEL	10, 149
ライブラリアンWLB	10, 150
THE SENTINEL	11, 129
タブコード対応エディタEDC-T	11, 130
THE SENTINEL	12, 133
STACK コンパイラ	12, 134

連載・シリーズ

知能機械概論 お茶目な計算機たち

第34回 1000bitプロセッサの世界	1, 106
第35回 パソコンキッズの自由な魂が踊り出す!	2, 92
第36回 ノスタルジアという病	3, 98
第37回 本は捨てられない	4, 156
第38回 次世代マイクロプロセッサ登場	5, 170
第39回 マックやめませんか、それとも人間	6, 162
第40回 人工知能の冒険	8, 158
第41回 超能力実験の成果	9, 140
第42回 続・超能力実験の成果<透視>	10, 162
第43回 超遊園地都市「ロココ町」の住人	11, 162

猫とコンピュータ

第43回 夢をセールス	1, 108
第44回 通信売りまくり	2, 94
第45回 自動ドアと初もうで	3, 100
第46回 PI! PI!! が多すぎる	4, 158
第47回 開け! ファイル	5, 172
第48回 FASTでなくちゃ	6, 164
第49回 ホットラインで	7, 134
第50回 サチャーでござる	8, 160
第51回 PTAは2度死ねない	9, 142
第52回 「文豪」が飛んだ	10, 164
第53回 風の日のロボコン	11, 164
第54回 マニアの砦にて	12, 166

X-OVER NIGHT

第1話 いろいろ選択の自由?	6, 161
第2話 本命は恐ろしい	7, 153
第3話 旅行あれこれ	8, 162
第4話 流行歌を追え!	9, 144
第5話 2つの人種?	10, 161
第6話 花博が終わって	11, 161
第7話 スキーは豪華に?	12, 169

大人のためのX68000

第0回 長い能書きでごめん	9, 38
第1回 住所録あれこれ	10, 60
第2回 大人は数字を弄ぶ	11, 58
第3回 スプレッドシートに精神	12, 63

X-BASICプログラミング調理実習

(7) ギターで遊ぼう	2, 111
(8) MMLで演奏に挑戦してみよう	3, 104
(9) エレベータのシミュレータ	4, 58
(10) エレベータのシミュレータ(2)	5, 162
(11) パズルゲームを作る(前)	6, 74
(12) パズルゲームを作る(後)	7, 108
(13) 超入門・ファイル処理	8, 107
(14) ファイルの魔術師fseek関数	9, 76
(15) カード型データベース(1)	11, 65
(16) カード型データベース(2)	12, 92

X68000マシン語プログラミング

Chapter_0A 「プロセス操作」という世界	1, 75
Chapter_0B ファイル管理の方法	2, 121
Chapter_0C デバイスドライバを作る(前)	4, 49
Chapter_0D デバイスドライバを作る(後)	5, 153
Chapter_0E 脱「入門者」のための身辺整理	6, 79
Chapter_0F マウスwithグラフィック	8, 113
Chapter_10 直接グラフィックを操作する	9, 81
Chapter_11 画像に変化を与える処理	10, 51
Chapter_12 グラフィックパターンの扱い方	11, 83
Chapter_13 C,X-BASICの関数を作成する	12, 97

C調言語講座PRO-68K

第18回 思考よ〜ん(その1)	1, 85
第19回 思考よ〜ん(その2)	2, 130
第20回 思考よ〜ん(その3)	3, 102
第21回 思考よ〜ん(その4)	4, 68
最終回 終わりにだからターミナルなのよ	6, 89

ようこそここへC言語

第1回	プログラムって何だろう	10, 46
第2回	変数って何だろう	11, 72
第3回	制御構造って何だろう	12, 142
PASCALプログラミングへの招待		
<1>	X68000にPASCALコンパイラを	6, 94
<2>	PASCALの特徴的な性格について	7, 114
<3>	PASCALのデータ型を見る	8, 121
<4>	PASCALの制御構造、関数および手続き	9, 71
<5>	演算子・式、インラインアセンブラ	11, 78
清水和人流プログラミング道場		
その1	アマグラマに花道を	9, 40
その2	ああ、ピアニスト	10, 124
シミュレーションプログラミング入門		
第1回	コンピュータシミュレーションの世界	12, 67
D&G・CGアニメーション講座		
<7>	脳みそどろどろフレームソース	1, 90
<8>	1990年運営基本方針発表!	2, 134
<9>	映像表現のテクニック	3, 118
<10>	春だ、4月だ、新歓だ!	4, 46
<11>	復活のCGアニメーション	7, 96
<12>	こんな表現、あんな表現	9, 104
<13>	CGA緊急レポート 夏だ、祭りだ、合宿だ!	10, 68
<14>	宇宙要塞CADの逆襲 その1	11, 118
ハードウェア工作入門		
予告編		6, 131
<1>	基本インタフェース回路 その1	7, 105
<2>	基本インタフェース回路 その2	8, 104
<3>	基本インタフェース回路 その3	9, 136
<4>	A/Dコンバータ その1	10, 65
<5>	A/Dコンバータ その2	11, 125
<6>	A/Dコンバータ その3	12, 82
マシン語カクテルin Z80's Bar		
第7回	ドライブに連れてって(1)	1, 113
第8回	ドライブに連れてって(2)	2, 105
第9回	ちょっと待ったコヘ!	3, 113
第10回	にじゅうまる、あげよー	4, 145
第11回	ライン文だべっちゃ!	5, 150
第12回	帰ってきたゼンジソフト	6, 145
第13回	電卓はビットシフトで	7, 117
第14回	楽な逆ポーランド?	8, 126
第15回	ハッシュでチェック	9, 152
第16回	ソーティングって?	11, 155
第17回	私はエディタ、原稿まだかな	12, 148
X1/turbo用ディスク管理プログラムINTEGRAL X1		
INTEGRAL X1		6, 134
ノーマルX1への対応		7, 137
トランジェントコマンドを作る		8, 92
X68000の画像を読み込むには		10, 140
外部コマンドの作成		11, 148
メニューによるファイル管理		12, 153
(て)のショートプロバート		
その5	お年玉(?)はユーティリティ	1, 102
その6	気分は数学ー!	2, 96
その7	掲載率10倍アップだよー!!	3, 126
その8	便利なsp_chk()	4, 72
その9	夜、見ないよーに	5, 146
その10	ちょっとへびい	6, 150
その11	前夜祭はすたっきだ!	7, 126
その12	祝! 1周年記念	8, 130
その13	なさない星?	9, 132
その14	ゲーム&ゲーム	10, 74
その15	テクニックは偉大なのだ!	12, 73
Oh!X LIVE in '90		
さよならを過ぎて(X1/turbo)		1, 97
RYDEEN (X68000)		1, 97
オーダインよりラウンド1のテーマ(X68000/MZ-2500)		2, 98
魔女の宅急便より 仕事はじめ(X1/turbo)		2, 98
ANGEL SMILE(X68000)		3, 88
パスワードリフトより Artistic Traps (X68000)		3, 90
ザ・スキームより PERPETUAL DARK! (X68000)		3, 91
エヴリデイ(X1/turbo)		3, 93
となりのトロよりねこバス(X1/turbo)		3, 95
バーニングフォース(X68000OPMD対応)		4, 77
TURBO OUTFUNより RUSH A DIFFICULTY (X68000)		5, 141

バレードしようよ(X1/turbo)	5, 141
ナイトアームズより3Dステージメインテーマ(X1/turbo)	6, 154
悪魔城伝説より Beginning(X68000)	6, 154
この木なんの木(X68000)	6, 154
夢幻戦士ヴァリスIIより SACRED SACRIFICE(X68000)	7, 121
トッカータとフーガ短調BWV565(X1/turbo)	7, 121
OMENS OF LOVE(X68000)	8, 134
ENDLESS RAIN(X1/turbo)	8, 134
ダートフォックスより Running up! (X68000MUSICDRV サンプル曲)	8, 134
風の谷のナウシカ(X68000)	9, 129
ラジオ体操第一(X1/turbo)	9, 129
Rise And Fall(X68000)	10, 118
NINJA WARRIORSより PARADOX (X68000)	10, 120
キュービー3分クッキングのテーマ(X68000)	10, 122
Marbel Land(X1/turbo)	10, 123
ビラミッドソーサリアンより GUSH(X68000)	11, 92
ザ・スキームより I'll save you all my justice(X68000)	11, 92
グラディウスIIIより Sand Storm(X68000)	12, 78
メタルサイトより Into the shadow (X68000)	12, 78
X68000CARD,FNC用カードゲーム	
X68000用カードゲーム支援関数	
CARD,FNC	5, 65
HEART・負けるが勝ち	8, 88
赤黒(SPEED)	9, 125
BLACK JACK	9, 125
ひとり占めTEN	10, 58
Rolling Stone	11, 62
COUPLE	12, 164
PC-E500テーブルトークRPGサポートシステム	
ポケコンでCARPGを	8, 102
マスター戦闘支援ツールCST	9, 156
シナリオ作成の支援	10, 78

機種別活用・プログラム

MZ-700	
Eyeland	2, 142
MZ-1500	
タコ釣りゲーム(ショート)	3, 127
GREEN(ショート)	6, 150
MZ-2000/2500	
αCで書く正規表現(特集)	1, 126
Round Around(ショート)	2, 97
RPG The Cave of Dalk	4, 121
Oh!X LIVE in '90	(→連載)
X1/turbo	
αCで書く正規表現(特集)	1, 126
MENU.bas(ショート)	1, 102
シミュレーションゲーム Super Battle	1, 110
Numbers(ショート)	2, 96
MIDIデータロード&セバ(特集)	3, 74
シミュレーションゲーム CRISIS in Tokyo	3, 108
ミサイルアタック(ショート)	4, 72
RPG The Cave of Dalk	4, 121
Z80's Bar番外編 通信によるファイル転送(特集)	5, 87
DIGMAN(ショート)	5, 146
STACKY(ショート)	7, 127
4096色→8色変換 Zの画像をX1で(特集)	8, 77
THE FANFAN(ショート)	8, 130
ばらんし(ショート)	10, 75
KURUPER(ショート)	12, 75
X1/turbo用ディスク管理プログラム INTEGRAL X1	(→連載)
Oh!X LIVE in '90	(→連載)
X1turbo	
電卓(ショート)	3, 128
Date Changer(ショート)	9, 132
X68000	
基本コマンド攻略法(特集)	1, 50
基礎から学ぶパッチファイル(特集)	1, 57
マジカル・ヒストリー・ツアー(特集)	1, 61
エイリアス主義のすすめ(特集)	1, 65
OS-9プログラミング教習所(特集)	1, 67
再帰大作戦(特集)	1, 118
GCCに見る最適化(特集)	1, 131

GMOVE,FNC For X-BASIC(ショート)	1, 102
YC分離とAD PCM(特集)	2, 50
ウォルシュアダマール変換を使う(特集)	2, 56
これが噂のPIC,R(特集)	2, 75
ゲームプログラム Gon Gon	2, 118
MIDI DRV, SYS(特集)	3, 38
なんでも鳴らせる OPMD,X(特集)	3, 43
LA音源用音色エディタ(特集)	3, 60
AI音源用音色エディタ(特集)	3, 71
MMLを楽譜データに(特集)	3, 78
エスケープオブメーズ(ショート)	3, 127
sp_chk() (ショート)	4, 73
X68000用ラジコンスティックの製作	5, 52
X68000用カードゲーム支援関数 CARD,FNC	5, 65
入門者のためのX-BASICの心得(特集)	5, 74
潜入! バグ対策24時間 XBASICはいま! (特集)	5, 76
X-BASICでゲームを作る 豪華版 SCRAMBLE(特集)	5, 80
CARD,FNCを活用する カードゲームを作ろう(特集)	5, 83
アルゴリズムを考える 拡大縮小処理の基本(特集)	5, 92
BASICで作るXCオブティマイザ(特集)	5, 97
空飛ぶDNAデモ(ショート)	5, 147
夜中にひとりで見てはいけないデモ(ショート)	5, 147
便利なツールあれこれ(特集)	6, 50
パズルゲーム Yet Another Column(特集)	6, 51
音楽演奏関係 OPMD & MUSICDRV(特集)	6, 52
X68000用GNU Cコンパイラ GCC Ver. 1.36.01(特集)	6, 54
ウイルス検出プログラム DOCTOR,R(特集)	6, 56
インテリジェントソースジェネレータ DIS,X(特集)	6, 58
対話型CGAシステム ANGEL(特集)	6, 62
SX-WINDOW & キャンバス,X用グラフィックデータコンバータ(特集)	6, 68
へびのぼーけん(ショート)	6, 150
基本装備とおまじない(特集)	7, 58
DOSコール&IOCSコールを使う(特集)	7, 63
マルチタスクへの挑戦(特集)	7, 67
ぱっくりあ,X(ショート)	7, 126
X68000用グラフィックツール紹介(特集)	8, 44
アンチエリアスとは?(特集)	8, 50
後処理によるジャギーの除去(特集)	8, 68
色間の補間と量子化(特集)	8, 72
X68000用画像回転プログラム XROT0,X	8, 81
かべくずし(ショート)	8, 130
我慢せずに使うWP,X(特集)	9, 51
ASK68K用辞書メンテナンスツール(前編)(特集)	9, 58
X68000用ハンディイメージスキャナアダプタの製作	9, 89
デジタルペインティングへの道(特集)	9, 113
なさない星(ショート)	9, 133
それ行け! ロケット(ショート)	10, 73
PIXYとX68000とMML FM音源のある部屋(特集)	10, 82
システム上でBGMをOPMファイルで遊ぶ(特集)	10, 84
音楽プログラムは怖くない(特集)	10, 86
MML記述テクニックYこそすべて(特集)	10, 90
多彩な表現をマクロにする ZMUSIC,FNC(特集)	10, 97
簡易OPMファイルジェネレータ OMUSIC,FNC(特集)	10, 113
ASK68K用辞書管理ユーティリティ(後編)	
辞書整備応用編	10, 128
とろける,X(ショート)	12, 73
XC ver. 2.0ガイドマップ(特集)	12, 106
ソースコードデバッガを使ってみよう(特集)	12, 112
線の下プリプロセッサ(特集)	12, 116
Cライブラリ活用の手引き(特集)	12, 121
XCにMAKEが付いてきた(特集)	12, 127
新しい音楽ドライバOPMDRV2,X(特集)	12, 129
XBAStoC CHECKER(特集)	12, 131
アナログジョイスティックの製作(特別企画)	12, 86
大人のためのX68000	(→連載)
X-BASICプログラミング調理実習	(→連載)
X68000マシン語プログラミング	(→連載)
C調言語講座PRO-68K	(→連載)
ようこそここへC言語	(→連載)
PASCALプログラミングへの招待	(→連載)
清水和人流プログラミング道場	(→連載)
D&G・CGアニメーション講座	(→連載)
ハードウェア工作入門	(→連載)
Oh!X LIVE in '90	(→連載)

X68000 CARD, FNC用カードゲーム……………(→連載)	
PC-E500	
PC-E500用テーブルトークRPGサポートシステム(→連載)	

イベント／ギャラリー

イベント

X68000ユーザー大集合!!	
シャープ「見・体・験フェア」in東京……………	2, 20
D&GAアマチュアCGAコンテスト……………	4, 25
シャープ見・体・験フェアin東京……………	6, 30
マイコンショウ'90 & 第70回ビジネスショウ……………	7, 27
エレクトロニクスショウ & データショウ……………	12, 38
Oh!X Graphic Gallery	
D&GA・CGA／再帰大作戦……………	1, 44
D&GA・CGA／画像圧縮へのアプローチ/Eyelarth……………	2, 22
D&GA・CGA……………	3, 32
D&GA・CG アニメーション……………	7, 40
2Dグラフィック続編/D&GA・CG アニメーション……………	9, 20
C-TRACE68 ver. 3, 0/CANVAS PRO-68K……………	10, 24
D&GA・CG アニメーション講座……………	11, 30
Oh!X Reader's ぎやらい	
「あけましておめでとー」のコーナー……………	3, 30
カラーイラスト大集合……………(→特別企画)	
THE USER'S WORKS	
ういっす2/GEM DROP……………	10, 23
DEMON SLAYER他……………	11, 29
X Turbo用グラフィクス……………	12, 39

製品紹介

ハードウェア

ハードディスク & プリンタ……………	1, 46
うわさの68040, ついに登場……………	4, 162
新機種登場!	
X68000 SUPER/EXPERT/EXPERT II/PRO II……………	5, 24
新製品レポート パソコンFAXアダプタ	
HALFAX9600/9600EX……………	5, 50
新製品紹介 いよいよ発売開始! X68000SUPER-HD……………	5, 154
新製品紹介	
ビデオボード/C compiler PRO-68K ver. 2, 0……………	9, 36

ソフトウェア

X68000の新しいビジュアル環境	
これがSX-WINDOWだ!……………	5, 121
追跡レポート これがSX-WINDOWだ(2)	
未来を語るアクセサリ……………	6, 98
新製品紹介 プログラミングユーザーの必須アイテム	
C compiler PRO-68K ver. 2, 0……………	10, 42

INFORMATION

ペンギン情報コーナー

X68000用LANボード CZ-6BLI(シャープ)……………	1, 162
カラー印刷可能な書院 WD-A900/A710(シャープ)……………	1, 162
デザインシミュレーションシステム CS-1(ユニテックジャパン)……………	1, 162
低価格MNP5モデム 通信ポコ(インテグランド)……………	1, 162
モデム新製品 MD24/96シリーズ(立石電機)……………	1, 162
プリンタパファ PRBシリーズ(アイ・オー・データ機器)……………	1, 163
ファミコン通信アダプター(マイクロコア)……………	1, 163
時計付き電子メモ PA-220(シャープ)……………	1, 163
OS-9000発表(マイクロウェア・システムズ)……………	1, 163
3次元グラフィックス(アーマット)……………	1, 163
低価格拡張スロット(システムショップハードソン)……………	1, 163
X68000用ディスプレイ CZ-604D/CU-14HD/CU-21HD/	
XV-100R(シャープ)……………	2, 154
X68000用メモリボード KGB-X68PRK(計測技研)……………	2, 154
X68000用MIDIボード SX68M(システムサコム)……………	2, 154
電子システム手帳 PA-8600(シャープ)……………	2, 154
パーソナルファクシミリUX-20(シャープ)……………	2, 155
3.5インチフロッピー MF/2HD 256(住友スリーエム)……………	2, 155
今日から電子手帳にしましょう(BNN)……………	2, 155
X68000環境ハンドブック(工学社)……………	2, 155
高機能・高速ワープロ WD-5600(シャープ)……………	3, 162
X68000用低価格拡張メモリ PIO-6BEI-A/2-M/4-M(ア	

イ・オー・データ機器)……………	3, 162
低価格自動プリンタ切り替え機 Auto Boy(八戸ファームウェアシステム)……………	3, 162
汎用的なFAXアダプタ HALFAX-9600(HAL研究所)……………	3, 162
巨大メモリのゲームマシン NEO-GEO(SNK)……………	3, 162
ハンディ無線機 TH-25G/45G(ケンウッド)……………	3, 163
携帯用マッサージ機とことん・HM-30(オムロン)……………	3, 163
MC68040出荷される(日本モトローラ)……………	3, 163
業務用プログラム注文制作(テラダ商電)……………	3, 163
電子手帳用パソコン通信(シャープ)……………	3, 163
ゲームセミナー参加者募集(任天堂・電通)……………	3, 163
TRONプロジェクト'88-89(パーソナルメディア)……………	3, 163
MC68030搭載のUNIXオフコン OA-220(シャープ)……………	4, 164
電子手帳用ICカード(シャープ)……………	4, 164
MNPクラス10の通信用モデム QX/2400t(ヒューコム)……………	4, 164
低価格ハンディコピー機 KX-261(松下電器産業)……………	4, 164
声の出る電子手帳 EX-3000(セイコー電子工業)……………	4, 164
車搭載用レーダー内蔵液晶テレビ N-9000(日本データ機器)……………	4, 165
親指大のビデオカメラ WV-KS102(松下電子工業)……………	4, 165
人の動きを感知するセンサライト FS-240(日立マクセル)……………	4, 165
「あいて」を打つマイク SAM-1(ソニー)……………	4, 165
ポケコンで制御トレーニング PMB-MC(太平洋工業)……………	4, 165
低価格コピーマシン Z-72(シャープ)……………	5, 174
カラー静止画TV電話 テレパシー LU-C10P(三菱電機)……………	5, 174
立体物をコピー・ダ・ピンチ(キングジム)……………	5, 174
手書き入力方式電子ノート IN-5000(キヤノン販売)……………	5, 174
個人情報管理PalmTop PTC-500(ソニー)……………	5, 174
Z80高速化される(シャープ)……………	5, 175
X68000とMS-DOSをリンク XIN/XOUT(データスペックジャパン)……………	5, 175
夢のクリスタルギャラリー(シャープ)……………	5, 175
CGAコンテストビデオ配布(D&GA)……………	5, 175
留守番録音FAX DA-7000(シャープ)……………	6, 166
携帯用CD-ROMシステム KV-M101(松下電器産業)……………	6, 166
世界最小のビデオコンポ GET'S(日本ビクター)……………	6, 166
ホームシアターテレビ 31C-E100(シャープ)……………	6, 166
自動的にCMカット HV-S95/F93/F92/D91(三菱電機)……………	6, 167
理系学生のためのポケコン PC-1490U(シャープ)……………	6, 167
アクセサリ新製品 Wikiシリーズ(日立マクセル)……………	6, 167
X68000の音をMIDIとミックス オーディオ・ミキシング・ケーブル(電波新聞社)……………	6, 167
ゲームソフトBGM新発売 ゲームボーイ グラフィティ(日本コロムビア)……………	6, 167
情報ツール「All in Note」AX286N-H2(シャープ)……………	7, 156
UNIXオフィスプロセッサ OA-120(シャープ)……………	7, 156
書院シリーズ & ニュー書院 WD-A620/A720/7000(シャープ)……………	7, 156
CCITT規格V, 42bis, 38400bps Multi modem V32L(コア)……………	7, 157
Z80完全上位互換16ビットCPU R800(アスキー)……………	7, 157
レーザービクアップ駆動回路のIC化 IR3C10/IR3C11(シャープ)……………	7, 157
OS-9/X68000用 SrcDbg V2, 0(マイクロウェア・システムズ)……………	7, 157
シャープ ワールドサッカー'90 in 東京……………	7, 157
スーパーアウトラインフォント内蔵 WD-A320/340(シャープ)……………	8, 164
X68000用ビデオボード CZ-6BVI(シャープ)……………	8, 164
X68000とMacをリンクMac版「XIN/XOUT」(電機本舗)……………	8, 164
電子手帳用プリンタ & 名刺管理カード CE-80P, PA-7C50/7C51(シャープ)……………	8, 165
32ビット浮動小数点DSP DSP96002(モトローラ)……………	8, 165
「X68000グッズショップ in Akihabara」ミナミ電気本館	
5階……………	8, 165
64KバイトRAM標準装備 PC-E550(シャープ)……………	9, 162
ハンディターミナル HC-70(セイコーエプソン)……………	9, 162
ノートワープロ WV-700(シャープ)……………	9, 162
ポケットモデム MD24FP4 II/5II(オムロン)……………	9, 162
電子手帳用ICカード PA-3C19/22~24/26~29(シャープ)……………	9, 163
X68000用アディンボード TP-68K/K-TRAM-2(国際データシステム)……………	9, 163
第2回 サイクロンCG大会……………	9, 163
高輝度液晶ビジョンXV-HI/HIZ(シャープ)……………	10, 166
ビデオレセプター XC-100P(シャープ)……………	10, 166
電子手帳で測量 PTS-IIIシリーズ(旭精密)……………	10, 166
電子手帳で過去帳管理 お寺さんカード(シャープ)……………	10, 167

X68000 COMPLETE BOOK(徳間書店)……………	10, 167
第7回 ホビーマイコンショウ……………	10, 167
第4回 全国(草の根)BBS大会……………	10, 167
Oh!X & Oh!FM バックナンバーフェア……………	10, 167
ハイパー電子システム手帳 PA-9500(シャープ)……………	11, 166
入力しやすい電子メモ PA-440/450(シャープ)……………	11, 166
競馬予想カード 七冠馬(日本ブレインウェア)……………	11, 166
高速仕様 XIN/XOUT アウトバーン(電機本舗)……………	11, 167
初期化済みフロッピーディスク MD/2HD(住友スリーエム)……………	11, 167
健康ソフトハウス物語, コンピュータよもやま話……………	11, 167
'90東京理工工学書展示即売会……………	11, 167
世界最大のコンピュータショッパ」&P新テクノランド……………	11, 167
TFTカラー液晶ディスプレイ搭載 AX386LC(シャープ)……………	12, 174
48/24ドットデュアルモードヘッド搭載 XP-2000(セイコーエプソン)……………	12, 174
マルチメディア対応TFTカラー液晶ディスプレイを開発(シャープ)……………	12, 174
ハンディステッカーマシン ステカ(ローランド ディー・ジュー)……………	12, 174
電子手帳用カード新3機種 PA-3C20/30/31(シャープ)……………	12, 175
32ビットIPU MC68340(モトローラ)……………	12, 175
ローランド主催 第3回 デスクトップ・ミュージック「力作」コンテスト……………	12, 175
FI情報 ネットで(フジミック)……………	12, 175
コンピュータ図書フェア 書泉グランデ5階……………	12, 175
FILS Oh!X 新刊書案内	
ハッカー英語辞典……………	1, 164
シミュレーション発想……………	1, 165
コピーって? ゼロックスですか……………	1, 165
IBM21世紀への挑戦……………	2, 156
コンピュータウイルス……………	2, 157
パソコンしな専科……………	2, 157
プログラマ・サバイバル・ガイド……………	3, 164
パソコンブックガイド……………	3, 165
科学を愛したサル……………	3, 165
タイムトラベラー2038年……………	4, 166
図解C言語構造化プログラミング作法……………	4, 167
ur ハイパー・アート……………	4, 167
映画: ブレードランナー……………	5, 176
エデンの西(上)……………	5, 177
BBSの歩き方……………	5, 177
コンピュータ社会が崩壊する日……………	6, 168
都市の書物……………	6, 169
ニューメディアの興亡……………	6, 169
情報の歴史……………	7, 158
作品としてのプログラム……………	7, 159
バックランドでつかまえて……………	7, 159
キューブリック・ミステリー……………	8, 166
エッシャーからの贈り物……………	8, 167
人は「無意識」の世界で何をしているか……………	8, 167
ロコ町……………	9, 164
数学オンチの諸君!……………	9, 165
パソコン少年のコスモロジー……………	9, 165
ライフゲームの宇宙……………	10, 168
コンピュータソフトウェアマニュアルの書き方……………	10, 169
ザ・ゲームカタログ'90……………	10, 169
コンピュータウイルス!……………	11, 168
ネットワーク・ベイビー……………	11, 169
メタマジック・ゲーム……………	11, 169
サイバテックノロジー……………	12, 176
心の社会……………	12, 177
WORLD ATLAS……………	12, 177

その他

Oh!X INDEX'90……………	12, 170
常設コーナー	
愛読者プレゼント	
ペンギン情報コーナー	
FILES Oh!X	
Oh!X質問箱	
STUDIO X	
編集室から/DRIVE ON/こめんなさいのコーナー/SHIFT BREA	
K/microOdyssey	
特別付録	
X68000 THE GAME SOFTWARE CATALOGUE (1月号)	
5"2HDディスク創刊8周年記念PRO-68K(6月号)	

NEW PRODUCTS

TFTカラー液晶ディスプレイ搭載 AX386LC シャープ

AX386LC



シャープは、鮮明表示の10インチTFTカラー液晶ディスプレイを搭載した32ビットラップトップ型AXパソコン「AX386LC」を発売する（12月15日予定）。

表示部には10インチTFT（薄膜トランジスタ）カラー液晶ディスプレイ（640×480ドット、64色中16色表示）を採用し、鮮やかなカラー表示が可能となっている。

本体のほうもクロック周波数20MHzの32ビットCPU 80386DX（ノーウェイト）、2Mバイトのメモリ、100Mバイトのハードディスク（平均アクセスタイム17ms）、16ビットのPC/ATバスを採用したハーフサイズ拡張スロット（AX仕様準拠）を1スロット、およびモデムなどが内蔵できる内部専用スロットを1スロットを搭載し、デスクトップマシンに相当する性能を実現している。

価格は1,490,000円（税別）。

〈問い合わせ先〉

シャープ(株) ☎03(260)1161, 06(621)1221

48/24ドットデュアルモードヘッド搭載 XP-2000 セイコーエプソン

セイコーエプソンではデュアルモードヘッドを搭載した48ピンドットマトリクス漢字プリンタ「XP-2000」を発売した。デュアルモードヘッドによって高品位印字実現のための48ドットモードと高速印字のための24ドットモードの切り換えが可能になっている。価格は196,000円（税別）。

また、同社ではターミナルプリンタの普及価格帯におけるラインアップを一新した。発売されたのは、シリアルインパクトプリンタ「VP-960/1600/2600」、熱転写プリンタ「AP-600/900」、インクジェットプリンタ「HG-4000」の計7機種。各製品とも従来機種に比べて低価格を実現しながら、印字スピードのアップなどの機能強化が図られている。

標準価格（すべて税別）

VP-960 116,000円

VP-1600 123,000円

VP-2600 143,000円

AP-600 62,800円

AP-900 92,800円

HG-4000 196,000円

〈問い合わせ先〉

セイコーエプソン(株) ☎0266(58)1705

エプソン販売(株) ☎03(377)7001

マルチメディア対応 TFTカラー液晶ディスプレイを開発 シャープ

シャープは独自の画像処理技術と新しい構想に基づくデジタル駆動システムの確立により、640×RGB×480ドット、1670万色表示可能の高速高精細マルチメディア対応10型および14型TFTカラー液晶ディスプレイを開発した。

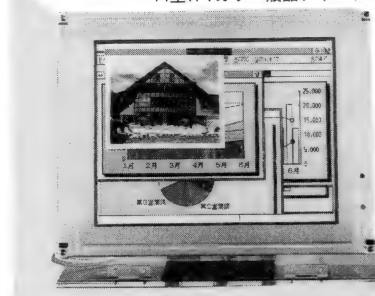
NTSCカラー入力やRGBセパレート入

力などのアナログ信号をデジタル階調に高速変換する画像処理部と、高速高集積のデジタル多階調ドライバLSIを搭載したTFT-LCDパネル部とで構成されており、オーディオビジュアル用の自然色動画表示とコンピュータコミュニケーション用の鮮明なマルチカラー表示の共用が可能で新しいディスプレイとして使用することができる。

〈問い合わせ先〉

シャープ(株) ☎03(260)1161, 06(621)1221

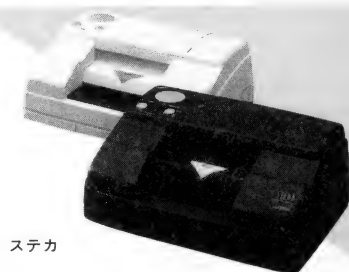
14型TFTカラー液晶ディスプレイ



ハンディステッカーマシン ステカ ローランドディー・ジー

ローランドディー・ジーでは、オリジナルステッカーの製作が誰にでも手軽にできるインテリジェントカッティングマシン「ステカ」を11月20日に発売する。

ステッカーにしたいロゴやイラスト、レタリングをハンディコピーの感覚でサッとナギリ、ステカシートと呼ばれる専用粘着シートを本体に差し込めば、フルオートで原稿のアウトラインがカッティングされる。ステカシートはカラフルなカラーバリエーションの屋外用、屋内用に加えて、アイロ



ステカ

ンプリント用シートもラインアップされ、Tシャツやトレーナーなどにも使用できる。

本体はグレーとホワイトの2色が用意されている。価格は75,000円(税別)。

<問い合わせ先>

ローランドディー.ジー.(株)

☎03(5687)1770

電子手帳用カード新3機種 PA-3C20/30/31 シャープ

シャープ電子手帳用ICカード3機種がソフト開発会社より新発売される。

・詰め将棋カード「寅詰」

新進気鋭の棋士、田中寅彦八段が選んだ挑戦問題200問を収録。

(株)スキップトラスト

標準価格7,500円(税別) 1月発売

・ロードランナー

人気アクションパズルゲーム「ロードランナー」の電子手帳版。全80ステージ。

(株)ナグザット

標準価格7,200円(税別) 発売中

・パズニック

あの「パズニック」が電子手帳で。128面+アレンジモードの“グラブニック”80面。

(株)タイター

標準価格5,000円(税別) 12月発売

<問い合わせ先>

シャープ(株) ☎03(260)1161, 06(621)1221



32ビットIPU MC68340 モトローラ

MC68340はM68000ファミリマイクロプロセッサとインテリジェント型データハンドリングペリフェラルを、シングルチップ上で集積した32ビットのインテグレートッ

ド・プロセッサ・ユニット (IPU) である。IPUはHCMOSプロセスを駆使してインプリメントされた完全スタティック動作可能なユニットで、低消費電力動作が可能。スタンバイモードも備えている。CPU32コアはMC68000およびMC68010とユーザーオブジェクトコードでコンパチブルであり、多くのMC68020の追加命令およびアドレッシングモードをも提供する。高速乗算、除算、およびシフト命令も備えているので多くのアプリケーションで性能を向上させることができる。例外処理機能も強化されておりリアルタイム動作で威力を発揮する。さらに、MC68020のパイプライン処理およびプリフェッチの概念も継承されている。

<問い合わせ先>

日本モトローラ(株) ☎0120-068030

INFORMATION

ローランド主催

第3回デスクトップ・ミュージック「力作」コンテスト

1988年に第1回を実施し、今回が3回目となるこの大会は、いわば“デスクトップミュージシャン”の日頃の成果を発表してもらうための場である。コンテストの結果は1991年2月に開催するコンピュータを使った音楽の祭典「デスクトップ・ミュージック・サウンド・パーティ」にて発表される予定。

○募集期間：'90年11月1日～'91年1月10日

○応募方法：有名パソコンショップ、楽器店、大会事務局に備えつけのコンテスト応募用紙を使用。3.5、あるいは5インチフロッピーディスクを添付。

○応募宛先：〒101 東京都千代田区神田須田町2-11 ローランド(株)内 第3回デスクトップ・ミュージック「力作」コンテスト事務局

○応募資格：プロ、アマ、年齢、性別など不問

○応募作品：5分以内。ジャンルは問わない。オリジナル/既成の楽曲どちらも可。ただし、既成の楽曲については作者とタイトルを明記

○賞品：グランプリには50万円相当、準グランプリには30万円相当のローランド製品

が進呈される。コンテストの審査機材は下記のとおり

パソコン	NEC PC-9800シリーズ シャープ X68000シリーズ 富士通 FM TOWNSシリーズ
音源モジュール	CM-64 (ミュージー郎) CM-32L (ミュージー郎Jr.) MT-32 (ミュージーくん) CM-32 P ローランドピアノ HPシリーズ ※ただし、パソコン本体の内蔵音源およびパソコンの拡張スロットに挿入する増設音源は使用しない。
使用ソフトウェア	自作ソフトウェアを含め、一切の制限はありません。
使用インタフェイス	ローランド MPUシリーズ シャープ CZ-68MI システムサコム SX-68M 富士通 FMT-602+FM-60-401 /FM-401に限ります。

<問い合わせ先>

コンテスト事務局 ☎03(251)2903

F1情報をネットで

パソコン通信「EYE-NET」の番組(メニュー)に「フジテレビF1情報(有料)」が新設される。いま人気のF1レースに関する情報を提供するこの番組では、予選第1戦、予選第2戦、決勝レースについてそれぞれ1本ずつ計3本のファイルをレースが行われた翌月曜日に掲載する。ピットウォークの裏話やレース予想なども掲載される予定。情報料金は1ファイルあたり250円。

<問い合わせ先>

フジミック ☎03(358)0591

コンピュータ図書フェア 書泉グランデ5階

書泉グランデでは「OSから広がるGUI、ネットワークの世界」と題したコンピュータ図書フェアを12月1日から来年の1月31日まで行う。

○OSの世界—UNIX、MS-DOS、OS/2など

○GUIの世界—X-Window, Windows, OSF/Motif, Open Lookなど

○ネットワークの世界—LAN, Netware, TCP/IPなど

○コンピュータの基本—アルゴリズム+データ構造, コンパイラ, C言語, PASCAL, Lispなど

以上の書籍を中心にコンピュータ関係の和書、洋書が取り揃えられている。

<問い合わせ先>

書泉グランデ ☎03(295)0011

FILES Oh!X

このインデックスは、タイトル、注記——筆者名、誌名、月号、ページで構成されています。何を食べてもおいしい季節になりました。いまのうちにいっぱい食べて、寒い冬に備えて力を蓄えておきましょう。

一般

▶ PIONEERS RETURN TO DALLAS SIGGRAPH '90

アメリカで毎年行なわれているコンピュータグラフィックの世界的祭典“SIGGRAPH”の今年の模様と作品を紹介。——編集部, LOGIN, 19号, 216-219pp.

▶ The News File

電子システム手帳のグレードアップ版「ハイパー電子システム手帳 (PA-9500)」。大型液晶ディスプレイタッチパネル、そのうえ記憶容量も64Kバイトというからうれし。——編集部, LOGIN, 20号, 38p.

▶ ハイテク地獄耳

大容量の関数ポケコンPC-E550, 450人の電話番号を記憶できる電子メモPA-450, プリント分離型ワープロWV-700を紹介。——編集部, POPCOM, 11月号, 131p.

▶ ビーター・モリニュー来日

ポピュラスの作者、ビーター・モリニュー氏の来日の模様と、ポピュラスの世界一決定戦をレポート。モリニュー氏の新作「パワーモンガー」も紹介。——編集部, コンピューク, 11月号, 94-97pp.

▶ NEW PRODUCTS

450人分の電話番号が記憶できる情報ツール、シャープの電子メモPA-450を紹介。——編集部, マイコンBASIS Magazine, 11月号, 94p.

▶ これから始めるコンピュータミュージック

MIDIを使ったコンピュータミュージックの特集。音源の選び方、曲データの加工法、プロへのインタビューなどを掲載。——編集部, ASCII, 11月号, 250-272pp.

▶ あなたが使うあなたのパソコン

初心者ソフトを使えるようになるまでに焦点を絞り、ソフトの選び方、勉強の仕方などを提示する。——見沢康・間島英之, ASAHIパソコン, 11月号, 14-25pp.

▶ ノート型パソコン活用大作戦

今話題の各種ノート型パソコンを、製品、ソフト、PDSの紹介なども交えて比較検討し、その環境について考える。——編集部, マイコン, 11月号, 106-127pp.

▶ PRIVATE EYE

ヘッドホン風にディスプレイを目の前に固定する超小型コンピュータ用表示装置「PRIVATE EYE」を紹介。——編集部, マイコン, 11月号, 251p.

▶ やまさんのアルゴリズム・ブック

リスト処理に関するアルゴリズムを考える。今回のテーマはガーベジコレクションについて。——やまさん, マイコン, 11月号, 300-304pp.

▶ サラウンド・キット製作記

秋月電子通商のサラウンドキットを製作し、X68000についてそのサウンドがどのように変化するかを試す。

——市原昌文, I/O, 11月号, 241-244pp.

▶ なんでもQ & A

All in Noteの再インストール方法, MAXLINKドライバの使用法に関する質問などに答える。——編集部, マイコン, 11月号, 400-401pp.

MZシリーズ

MZ-1500 (MZ-5Z001)

▶ あの世界があるなら

お化けのあなたは、生きている普通の人にはイタズラをして暮らしていた。イタズラに飽きたあなたは天国の道へと向かっていった。スクロールゲーム。——BLUE POW, マイコンBASIS Magazine, 11月号, 123-124pp.

MZ-2500 (BASIC-M25)

▶ 移植版Stone of Theory

画面のブロックをすべて消す。名作アクションパズルゲームの移植版。——CRYSTAL Soft, マイコンBASIS Magazine, 11月号, 125-126pp.

X1/turbo/Z

X1シリーズ

▶ PONA

重力反転180度。悪い卵をやっつける。——四杉誠一, マイコンBASIS Magazine, 11月号, 155-156pp.

▶ TOTOLIS

5, 4, 3, 2, 1で消える。パズルゲーム。——石塚文雄, マイコンBASIS Magazine, 11月号, 157-158pp.

▶ ネオ投稿プログラムコーナー

モニタを改造してチェックサムを逐次表示させるソフト「MコマンドでSUMが付きます」を掲載。——じよるじよる倶楽部 (ソフト投稿者: 赤岩秀明), マイコン, 11月号, 210-216pp.

▶ X1+FM音源ボード (要NEW FM音源ドライバ)

▶ Knight Arms〜ロード中のテーマ〜

X68000アクションゲーム「ナイトアームズ」のミュージックをX1で。——KAZ, マイコンBASIS Magazine, 11月号, 191-192pp.

X1 turboシリーズ

▶ 最新ゲーム徹底解剖!!

ナポレオンの生涯がモデルのゲーム, ランベルールを徹底解剖。——編集部, LOGIN, 20号, 146-149pp.

▶ 読上公開質問状

CZ-601DをX1turboシリーズにつなぐことができるか、などの質問に答えている。——多田太郎, マイコンBASIS Magazine, 11月号, 91p.

▶ BLUPPU

2種類のゲームが楽しめる, パズルゲーム。——中西

参考文献

I/O 工学社
ASAHIパソコン 朝日新聞社
ASCII アスキー
コンピューク 角川書店
テクノポリス 徳間書店
ポケコンジャーナル 工学社
POPCOM 小学館
マイコン 電波新聞社
マイコンBASIS Magazine 電波新聞社
LOGIN アスキー

新刊書案内



NTT版の贈る「未来はバラ色サイバー色」シリーズの最新刊である、とてもいい本になるくらい啓蒙色の強い本を出しているNTT出版であり、「サイバーテクノロジー」もそんな香りをたたえている。本書も8人の筆者の原稿+監修者の月尾氏と筆者との対談という形式で、テクノロジーについて述べていく。内容はアークヒルズで行われている「アーク都市塾」の要約であり、編集は編集工学研究所で装丁は戸田ツトムで刊行はNTT出版となれば目を通さないわけにはいかない。

情報産業をテーマに西和彦氏や松井隼氏 (びあ総合研究所社長) が語り、アーティフィシャルリ

アリティをテーマに、武光裕氏がシミュレーションについて、笹田剛氏が都市計画とCGAについて、河口洋一郎氏がCGアートについて語る。最後は情報編集をテーマに、電撃遊戯大全の著者である石原恒和氏と編集工学研究所の松岡正剛が語る。コンピュータに代表されるテクノロジーを使って新しい試みをしている人たちの考えというのは非常に興味深く、旧態然とした社会の端で興っている文化に触れるには格好の書だ。 (K)

サイバーテクノロジー 月尾嘉男ほか8人著 月尾嘉男監修 NTT出版 ☎03(435)1212 A5版 214ページ 2,400円

弘幸, マイコンBASIC Magazine, 11月号, 159-160pp.

X68000

▶NEW SOFT

新着ゲーム, サイバリオ, ジェミニウイング, アクシス〜FZ戦記〜, ナイアス, Misty Vol.5を紹介。——編集部, LOGIN, 19号, 14-29pp.

▶最新ゲーム徹底解剖!!

対戦版シムシティに挑戦。シムシティは、本来コンピュータと対戦するゲームであるが、遊び方にひと工夫加えて人間対人間のルールを考えた。ほかにウルティマVの攻略法, X68000オリジナルRPGのラグーンを紹介。——編集部, LOGIN, 19号, 136-143, 166-167pp.

▶X68000新聞

熱血高校ドッジボール部サッカー編, アクシス〜FZ戦記〜, 遊撃王II, エメラルドドラゴンなどのほか, 新作の情報紹介。——編集部, LOGIN, 19号, 226-231pp.

▶NEW SOFT

アクシス〜FZ戦記〜の攻略法のほか, 発売予定のニューラル・ギア, ハイドライド3SV, エメラルドドラゴンを紹介。——編集部, LOGIN, 20号, 16-26pp.

▶最新ゲーム徹底解剖!!

新着RPGラグーンの攻略法第2回。——編集部, LOGIN, 20号, 154-157pp.

▶X68000新聞

待望のC compiler PRO-68K ver2.0のほか, 新着ゲーム“アトミックロボキッド”“パロディウスだ!”“機甲師団”“ハイドライド3SV”などを紹介。——編集部, LOGIN, 20号, 252-255pp.

▶GAMING WORLD

サイバリオ, 熱血高校ドッジボール部サッカー編の紹介。——編集部, テクノポリス, 11月号, 18-19pp.

▶新作ゲーム先取り Soft Flash

“パロディウスだ!”“アトミックロボキッド”“F15 ストライクイーグルII”など, 発売予定のゲームを紹介。——編集部, テクノポリス, 11月号, 29p.

▶攻略おすすめゲーム

アクションRPG“ラグーン”を攻略。——編集部, テクノポリス, 11月号, 56-58pp.

▶おにおんのアルゴリズムを見切ったぞ!!

カードゲームのアルゴリズムを解説。サンプルとしてX-BASICでのばねきゲームを掲載。——編集部, テクノポリス, 11月号, 118-122pp.

▶ゲームがオレを呼んでいる!

X68000オリジナルのアクションRPG, ラグーンの攻略法。——たかはび, POPCOM, 11月号, 68-71pp.

▶WE ARE THE X68000 WORLD

熱血高校ドッジボール部サッカー編, ナイアス, アク

シス〜FZ戦記〜, サイバリオ, 遊撃王II, ハイドライド3SVを紹介。——編集部, POPCOM, 11月号, 88-91pp.

▶ミュージック・パビリオン

“働く男(ユニコーン)”のミュージックプログラム。——編集部, POPCOM, 11月号, 167-170pp.

▶X68000SPIRITS

期待の“パロディウスだ!”をはじめ, 熱血高校ドッジボール部サッカー編, ハイドライド3SVを紹介。——編集部, コンプティーク, 11月号, 246-247pp.

▶誌上公開質問状

CZ-603Dにスピーカーをつなぐことができるか? 付属ワープロの「改行幅表示」とは何か? ——多田太郎, マイコンBASIC Magazine, 11月号, 92p.

▶NEW PRODUCTS

X-BASICをコンパイルする際に威力を発揮するデバッキングツール「XBASToC CHECKER」を紹介。——編集部, マイコンBASIC Magazine, 11月号, 94-95pp.

▶Mars Knows!

びしばし弾を撃って敵をやっつけていだけ! シューティングゲーム。——高橋秀之, マイコンBASIC Magazine, 11月号, 161-163pp.

▶爆風ポスト2

爆弾を使って手紙をポストに入れるゲーム。——まてりある, マイコンBASIC Magazine, 11月号, 164-166pp.

▶トリオ ザ バンチ

デタイーストの同名のゲームミュージックプログラム。——石田勇, マイコンBASIC Magazine, 11月号, 193-194pp.

▶チャレンジ! PDS & 同人ソフト

PDSや同人ソフトを紹介する新コーナー。今月はX6800用PDS“OH! HAJIKI FINAL”“仙人ゲーム”を紹介。——佐久間亮介・やんま, マイコンBASIC Magazine, 11月号, 234-235pp.

▶AVプログラミング講座

スプライトプログラミング実践編。実際にスプライトを動かすテクニックとスプライトの衝突判定について解説する。——仲田津宏, ASCII, 11月号, 321-328pp.

▶AV STRASSE

C compiler PRO-68K ver2.0とXBASToC CHECKER PRO-68Kの2本の内容を紹介している。——編集部, ASCII, 11月号, 361-364pp.

▶個性派パソコンシリーズ

「キミはとってもオシャレな遊び人」と銘打ってX68000の概要とその性格について紹介する。鳥居勉氏のインタビューも同時に掲載。——荻窪圭, ASAHIパソコン, 11月号, 116-121pp.

▶GAME REVIEW

遊撃王II, ラグーン, サイバリオの3本についての評価記事。——MUNEPI 桃・桃子・あゆかわさつみ, マ

イコン, 11月号, 193-203pp.

▶なんでもQ & A

XBASToC CHECKERとは何か, プリンタのタイムアウトに関する質問, シャープの新型ディスプレイの仕様を掲載。——編集部, マイコン, 11月号, 398-399pp.

▶GAME BOX

ポピュラスプロミストランド, 機甲師団, ジェミニウイングを紹介——市原昌文・吉沢正敏, I/O, 11月号, 130-132pp.

▶SQUARE-400

ボード上の駒をすべて消し去る知性と気力のパズルゲーム。——佐藤敏孝, I/O, 11月号, 173-175pp.

▶COPYX & CTRL

拡張版COPY命令と, FDDイジェクト制御のキーを定義するプログラム。——(は), I/O, 11月号, 176-184pp.

▶SOFT BOX

バージョンアップされたC compiler PRO-68Kを, 変更点を主体に解説。——L & M, I/O, 11月号, 205-207pp.

▶SCASM

PC-E500/550/1480U/1490UのCPU, SC62015のクロスアセンブラ。データ転送はRS-232Cを介して行う。——小笠原博之, I/O, 11月号, 161-172pp.

ポケコン

PC-E500

▶MonMon & Pikoron

貧乏なモンモンがピコロンの宝を盗む。アクションゲーム。——小川章, マイコンBASIC Magazine, 11月号, 168p.

▶ガンバレ セイビン2

整備員さんのパズルゲーム。——町野稔, マイコンBASIC Magazine, 11月号, 169p.

PC-E500/E550/1480U/1490U

▶金融商品損得計算プログラム

利息や複利のデータにそって商品の利率と利回りを算出する。——東浦丞示, ポケコンジャーナル, 11号, 92-93pp.

▶WOT

相手より先に4つの石を集めろ! アイテムや武器の豊富な対戦型ゲーム。——Itom, ポケコンジャーナル, 11号, 67-75pp.

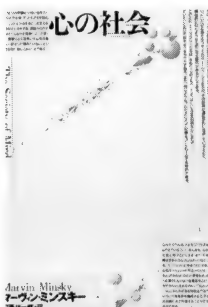
▶北京

“上海”風味の神経衰弱ゲーム。——CUE, ポケコンジャーナル, 11号, 76-77pp.

PC-1350

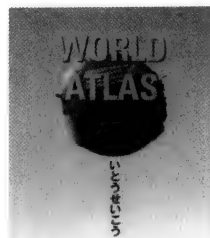
▶DRAGON WARRIOR

勇者に奪われた洞窟を奪い返せ! 龍が主人公のRPG。——せとけん, ポケコンジャーナル, 11号, 88-90pp.



心の社会

皆さまお馴染みミンスキー教授の, 邦訳が待たれていた「心の社会」である。「心の社会」というのは「心がたくさん小さなプロセスからできているという考え方」のこと。決して情報処理や人工知能の本ではないし, 心理学や社会学の本でもない。心というものに理系なアプローチをしたミンスキー教授の集大成だ。500ページ以上の大作だが, 専門用語に頼る部分もなく, 読みやすいレイアウトなので非常にありがたい。名著。(K) マービン・ミンスキー著 安西祐一郎訳 産業図書 03(261)7821 A5版 574ページ 3,800円



WORLD ATLAS

マガジンハウスのHot Dog Pressに連載されていたいとうせいこう氏のWORLD ATLASがやっと単行本になった。紙ではなく精神にマッピングされた世界地図とA〜Zまでのあやしい辞書形式エッセイ。値段が1,990円というのもあやしい。面白いのは氏のものの見方である。ぬぼーっと生きていては気づかないところ気づかせてくれる, といった感じだ。言葉って面白いと思う。この文が気に入らない人以外にはお薦め。(K) いとうせいこう著 太田出版 03(359)6262 四六版 260ページ 1,990円



最近、フォーマット済みのフロッピーをよく買ってくるのですがせっかくフォーマットしてあるのにもう一度X68000でフォーマットしなさいとSYS B:やCOPY COMMAND.X B:してもHuman68kが立ち上がるディスクが作れません。これってなんでなんですか？

長崎県 楠井 富男



「MS-DOSデータフォーマット済み」、「できるFD98初期化済」、「PC-98フロッピーDOSフォーマット」、「MS-DOS FORMATTED FOR PC-98」……最近MS-DOS用フォーマット済みフロッピーディスクがいろいろなメーカーから出ていますね、私も手元にあるフロッピーだけでこんなに種類がありました。なかにはPC-9801用でありながら対応機種にX68000の名前を載せX68000の写真（なぜかキーボードとディスプレイだけなんだけど）がパッケージに印刷されているものまであって私のようなものぐさな人間には天の恵みといえましょう（そんなごたいそうなもんかい）。

ご存じのようにPC-9801用MS-DOSでフォーマットされた5インチ2HDはHuman68kと同じく1024バイト/セクタ、8セクタ/トラック、154トラックという設定でフォーマットされていますので、そのままHuman68kのデータディスクとして使うことができるわけです。しかし残念なことに、どれも起動用のディスクとして使うためにはSYS.XでHuman68kを転送するだけではだめで、もう一度フォーマットしなさいしなければならないのですね。せっかくフォーマットしなくてすむと思って買ったのに……、なぜこんなことになるんでしょう？

ここではX68000が電源を入れられてCOMMAND.Xが立ち上がるまでを見ていきましょう。

まず、X68000の電源が入れられます。で、いきなりHumanが起動する……わけではありません。まず、電源が入った時点では68000MPUにスタックアドレスと最初に行するプログラムの入っている番地を入れてまず68000を動かさなくていけないのです（どうだめまいがしてきただろう）。

で、68000が動くのですが、そのときにプログラムの番地を入れましたね。つまりここではまだディスクから読み込んで動き

出すのではなくX68000上にプログラムがなくてはいけません。X68000の場合はROM上に動き出すためのプログラムが載っています。これをブートストラップローダといいます。

で、ブートストラップローダにはなにが書かれているかというと、諸々の初期化を行ったあとに“ディスクの先頭、トラック0サーフェス0セクタ1からの1セクタ分のプログラムを\$2000に読み込みそのプログラムを実行せよ”と書かれています。

いよいよこのプログラムによってディスク上のプログラムが読めます。ディスク上の最初に動くこの部分はIPLと呼ばれるものでこれがHuman.sysをディスク上から探してメモリ上に引っ張ってきて実行するという役目を持っているのです。そしていよいよHuman.sysが動き、command.xが動く……という構造になっています。

ではいったいフォーマット済みのディスクではなにがいけないのでしょうか？

それはIPLなのです。というのはFormatしたときに書き込まれてしまうので他機種でフォーマットしてしまうとX68000用のIPLが書き込まれないのでこのようなことになってしまうのです。つまり立ち上がらなかった原因はHumanのIPLが入ってなかったからだった、というわけなのです（ちなみに市販のフォーマット済みディスクでは0しか書き込まれていなかったり、PC-9801用のIPLが書き込まれていた

リスト1

```

1:      .include      doscall.mac
2:      .include      iocall.mac
3:      .text
4:      .even
5:
6: gosuper:
7:      clr.l          a1
8:      locs           _B_SUPER
9:      move.l         d0,a6
10:
11: InILDI:
12:      move.b         #16,d0
13:      lea            plttbl,a1
14:      move.l         #0e82200,a0
15:
16: movdat:
17:      move.w         (a1)+,(a0)+
18:      subq.b         #1,d0
19:      movdat         bne
20:
21:      bsr            DrawBar
22:
23: extdos:
24:      move.l         a6,a1
25:      locs           _B_SUPER
26:      lods            _EXIT
27:
28: DrawBar:
29: *Draw Color Bar
30:      move.l         #0e8000,a0
31:      clr.b          d0
32:      drwlop:        bsr
33:      lea            4(a0),a0
34:      addq.b         #1,d0
35:      cmp.b          #16,d0
36:      bne            drwlop
37:      rts
38:
39: drwvrm:            move.l         d0-d2/a0,-(sp)
40:      move.b         d0,d1
41:      move.b         #4,d2
42:      loptxt:        move.b         d1,d0
43:      and.b          #1,d0
44:      beq            ndtxt

```

り謎のデータが書き込まれていたりする)。

新たにX68000用のIPLを登録するにはディスクの再フォーマットが必要です。といっても、全体を初期化する必要はなく、

format /c

(FATとディレクトリだけの初期化)では時間もかからずIPLもきっちり書き込まれますのでお手軽にIPLを組み込みたいという人はこの/Cスイッチを使うのがいいでしょう。ただ、当然ですがディスク上のプログラムは消えますから使うときには気をつけてください。



X68000のカタログなどを見ると、テキスト画面は1024×1024（表示768×512）ドット、65536色中16

色となっているはずなのにいくらやってもテキストでは4色しか出ない。これはサギだ！

岡山県 倉本 仁



……いきなりサギといい切ってしまうのも凄い話ですが……。もちろんX68000はハード的にはカタ

ログに書かれているとおりきっちり16色出すことができます。ただHuman68kではそのように使っていないというだけの話なのです。

ではその隠れてしまった12色はどこにいったのでしょうか。実はこれパレットを細工することによって毎度お馴染みのソフトウェアキーボードとマウスカーソルに使われているのです。

で、プログラマーズマニュアルの図にあ

```

45:      bsr            writvrm
46:      ndtxt:         lsr.b          #1,d1
47:      add.l          #20000,a0
48:      subq.b         #1,d2
49:      bne            loptxt
50:      move.l         (sp)+,d0-d2/a0
51:      rts
52:
53: writvrm:
54:      move.l         d0/a0,-(sp)
55:      move.b         #24,d0
56:
57: loopBar:
58:      move.l         #0ffffff,(a0)
59:      lea            80(a0),a0
60:      subq.b         #1,d0
61:      bne            loopBar
62:
63:      move.l         (sp)+,a0/d0
64:      rts
65:
66:      .even
67:      .data
68:
69: plttbl:
70:      dc.w           00000 00000 00000 0      ; 0
71:      dc.w           00000 00000 10000 0      ; 1
72:      dc.w           00000 10000 00000 0      ; 2
73:      dc.w           00000 10000 10000 0      ; 3
74:      dc.w           10000 00000 00000 0      ; 4
75:      dc.w           10000 00000 10000 0      ; 5
76:      dc.w           10000 10000 00000 0      ; 6
77:      dc.w           10000 10000 10000 0      ; 7
78:      dc.w           00000 00000 00000 0      ; 8
79:      dc.w           00000 00000 11111 0      ; 9
80:      dc.w           00000 11111 00000 0      ; 10
81:      dc.w           00000 11111 11111 0      ; 11
82:      dc.w           11111 00000 00000 0      ; 12
83:      dc.w           11111 00000 11111 0      ; 13
84:      dc.w           11111 11111 00000 0      ; 14
85:      dc.w           11111 11111 11111 0      ; 15
86:
87:
88:      .end

```


るようにテキスト画面は文字のパターンを描くところが4枚あります。で、その左下に“各ビットごとのテキストパレットにアドレス”と書いてありますね。

この図ではよくわからないかもしれませんがね（シャープさんごめんなさい）。ここでいっていることは、簡単に説明すると文字のパターンが、

T3に描かれていたら8を

T2に描かれていたら4を

T1に描かれていたら2を

T0に描かれていたら1を

それぞれ足した数のパレットの色を出す、というふうになっているわけです。

つまり、T3とT0に文字のパターンが描かれていたら、

$8 + 1 = 9$

で9番のパレットにある絵の具を使って文字のパターンを出す……と考えてもらえばいいのです。

さて、テキストの文字とマウスカーソルなのですが、Human68kではT0とT1に文字が描かれT2、T3にマウスカーソルやソフトキーボードが描かれるようになっています。さっきの考え方でいくと文字を描いた上にソフトウェアキーボードを描くと文字とキーボードが重なった部分は変な色で表示されてしまうのではないか、と思うでしょう。しかしそこはパレットの魔術。実はHumanではパレットが、

0 黒

1 黄

2 青

3 白

4～7 マウスカーソルの枠の色

8～15 マウスカーソルの地の色

と設定されているのです。これならばたとえば白い文字の上にマウスカーソルの枠があっても、

$T0 + T1 + T2 = 7$

（マウスカーソルの枠の色）

ということになって文字の上にマウスカーソルが乗っているという重ねあわせが簡単にできてしまうのですね。

というわけで、IOCSではサポートされていませんがマシン語で直接I/OをいじってパレットのI/Oに16色分色を設定してやればちゃんと16色出せるんです。調子にノリついでにサンプルプログラムまでアセンブラで作ってしまいました。参考にしてく

ださいな。ちなみに、

リストの上段 ……パレットの設定

中段 ……カラーバーの描き込み

下段 ……パレット用の色データ

になっています（色のデータはBRGとテキストVRAMの対応がそのままになるようになっていきます）。（古村 聡）



初心者なもんで10月号83ページに載ってる毎日（毎曜日）違う曲を鳴らす方法がわかりません。チェックするっていうのとチェック結果をエラーコードで返すっていうのがわかんないんですが。できればサンプルリストを紹介してほしい……。お願いします。

福岡県 梶谷 太郎



チェックするというのは、今日が何曜日かということ調べることで、チェック結果をエラーコードで返すというのがわかりづらいかもかもしれません。プログラムはOSから呼び出されてなにかの処理をして戻ります。その際に、正常に終了したかどうかをOSに知らせるのです。この終了コードのうち0～255までの部分は起動したユーザープログラムで利用することが許されています。

終了日曜日を0、月曜日を1……と数字に置き換えて考えると、0から6ですべての曜日を表せますよね。この数値をエラーコードとして扱って、バッチプログラムでプロセス終了コードに応じた処理をさせる、ってことです。Human68kユーザーズマニュアルのCOMMAND.Xおよびバッチ処理コマンドのIFの項を参照してください。

このプロセス終了コードを指定するもっとも簡単な方法はC言語プログラムなどでメイン関数の戻り値として指定してやることでしょう。ここではBASICで書いた場合を例に挙げます（要Cコンパイラ）。

曜日を得る処理（day.bas）は次のようになります。

10 int i

20 str week = “日月火水木金土”

30 i = instr(1, week, day\$) * 2

40 end

きわめて簡単ですね。このままではなんにも出力しませんが、かまわずBC.XにかけてC言語プログラムにします。エディタで変換したプログラムを読み込み、“b_init()”という行を削って最後の“b_exit(0)”

の部分で“_exit(i)”に書き換えて、

cc day.c /W

でコンパイルしてください。できあがったday.xを、

echo off

day

if exitcode 1 copy sun.opm opm

if exitcode 2 copy mon.opm opm

:

if exitcode 7 copy sat.opm opm

echo on

のようなバッチファイルで利用すればできあがりです。

* * *

続いて先月のアフターケアです。先月号でワープロの文書を復活する方法を紹介しましたが、Cコンパイラのバージョン2についてくるデバッグはバージョンが2.00となっていてプログラムサイズも大きくなったので先月号の説明のとおりにしてもうまく動きません。

それから、先月のプログラムはデバッグのバージョン1.01の使用を前提にしているのですが、システムの使用状況によっては、説明どおりにやってもうまくできない場合があります。そんなときはデバッグ（バージョン不問）でサーチする方法が効果的ですので、うまくいかなかったら、とにかくサーチしてみてください。先頭番地さえわかればこっちのもんです。（影山 裕昭）

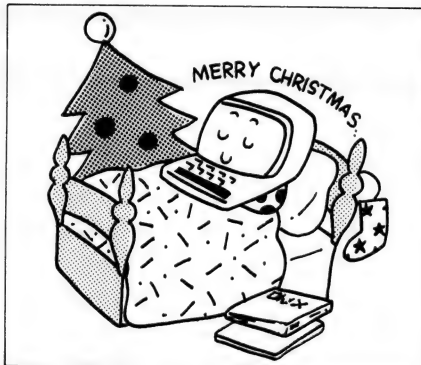
質問にお答えします

日ごろ疑問に思っていること、どんなことでも結構です。どんどんお便りください。難問、奇問、編集室が総力を上げてお答えいたします。ただし、お寄せいただいているものの中には、マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限、マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名、システム構成、必要なら図も入れてこと細かに書いてください。また、返信用切手同封の質問をよく受けますが、原則として、質問には本誌上でお答えすることになっていますのでご了承ください。なお、質問の内容について、直接問い合わせることもありますので、電話番号も明記してくださいね。
宛先：〒108 東京都港区高輪2-19-13

NS高輪ビル

ソフトバンク株式会社出版部

「Oh!X質問箱」係



FROM READERS TO THE EDITOR

先月は実際は暑いのに涼しいとか書いたけれど、今月は本当に涼しいぞ。どうだ、まいったか。でも、こういう季節は風邪

が流行るから気をつけなくては。そのためには温かくして寝るのがいちばん。さあ、さっさと寝よ寝よ。

◆「マシン語カクテル in Z80's bar」は、どうしちゃったんですか？ あのコーナーの登場人物みんなでワイワイやっているのが読んでいても面白くて気に入ってたんですけど終わっちゃったんですか？ それならすごく残念です……。

澤田 光彦(18)北海道

いやいや、そう早合点しないでください。10月号はお休みしただけです。どうぞご心配なく。

◆来月、防衛大の入試を受けに行きます。ところで、Oh!X編集部には防衛大出身の人はいますか？ (いたらスゴイですね)

川田 剛(18)大阪府

いません。そもそも、みんな身体弱いですから。ゴホゴホ。

◆1990年5月より買っています。X68000を買ったのはもっと前だったのですが、それまでは忙しくて触るひまがなかったのです。ところで、アンケートハガキにある「好きな石」とは？ なにか意味があるのでしょうか。とりあえず好きな鉱石を書いておきました(黄鉄鉱の結晶、真四角のが好き)。この石を見た時は天然物とは思えないほどきれいな正四面体をしていて、しばらく見入ってしまいました。IC(石)でいえばMC68000系(特に68040?)かな。X68000にもっと速くなってほしいし、「ダイヤ」もいいかな、硬い絶縁物だし。

城戸 吉巳(25)滋賀県

「ダイヤ」はいいですね。給料3カ月分とかいわれると困るけど。

◆私はパソコンのことはほとんどわかりません。だから、好きな石はなに？ という質問に真剣に「安山岩と花崗岩かなあー」と答えてしまいました。主人に話さず黙ってアンケートハガキを出していたら、私は間違いなく質問のところにマヌケな答えを書いていたでしょう。それではこれからがんばってください。

森本 幸子(24)千葉県

だいじょうぶ、「ウケ狙い」のつもりでICの名前を書いた人がたくさんいましたから。

◆「あなたの好きな石は何ですか?」、……よく考えるとはいません。1979年頃、8080Aで自作を始めたので、「思い出の石」というのはたくさんあるが、いまはハードを一から作ることはありえないので石の好き嫌いをいうレベルではない。CPUは68系が好きだったのでX68000を選んだのだ。

「私の思い出の石」

一瞬にして燃えた1個6,000円くらいだったVDGの6847。大容量メモリを作るつもりで大量購入したら2114が安く出回りに使わずにオクラ入りした1個1,200円くらいのSRAM2112。書き込み器の失敗でダメになったEPROM2708。

平山 謙司(40)福岡県

石に歴史あり、ですね。

◆10月号は音楽特集ということでLIVE in '90がいままでにして気合いの入ったいい曲があってよかった。自分はこのコーナー命の人間なのでどんどんいい曲が載るのを期待しています。ちなみに、いままでの曲の中でいちばんよかったのは「RYDEEN」です。この曲は感動しました。

城田 雅弘(17)群馬県

「RYDEEN」は最初の馬の足音がいいですよ。えっ、それはなかった?

◆先日、バンド仲間のライブを見に行き、とび入りで歌わされた。ひさしぶりのスポットのせいか右目がとても痛くなった。そして、曲の後半、フレーズのちょうど盛り上がりのあたりで涙がポトリ……。突然のことに客席もそのバンド仲間もびくりましたのだと思う。なんともいえないキンチョーが私のほうに伝わってきた。とっさに、「ちょっとと思うところがあって。ごめんなさい……」などとしっかりとコメントしてステージを降りた。あの雰囲気じゃ、まさか本当のことなど言えないよネ。それにしても皆さん、女の涙にはご用心ですよ!!

矢吹 準子(24)福島県

よくとっさにそんなコメントが出てきましたね。「ちょっと玉ねぎが目……」とかいったらさぞかし場がシラけただろうなあ。

◆会社の人に「X68000がほしいの」といったら、「PC-9801のほうが絶対にいいよ〜」といわれてしまった。そこで、「だって、PC-9801ってばカッコワルイんですもん」といったら、返す言葉もなく引き下がった。X68000がPC-9801に勝ったぞ。 田中 泰代(19)神奈川県

返す言葉がなかったんじゃなくて、あきれてしまったのかも。

◆本棚にOh!Xが5冊並んでいます(そのとりにミュージアの画集が……)。先日、なんとか前期試験も終わり、大型台風のお陰で遅れたOh!Xを読んでいると、父が「嫁入り道具のパソコンは何がいいか」といいました。しばらく何も言えない私。まだちゅうはちだよおう。でも、「X68000 HYPER-HD」(笑)あたりかな……なんて思うこともあります。次の新作は一体何だろう？

岩瀬 貴代美(18)福岡県

嫁入り道具にパソコンも買ってもらえるとはなんて幸せ。でも、結婚するころには次のX68000どころか、次の次のX68000とかが出てたりして。

◆今日は(9月17日)朝からいいことしたんだぜえ。一日一善ってやつだな。駅前で傘がなくて困っている高2の女の子と一緒に学校まで行ってあげたんだ。「どお?」って聞いたら、「すいませ〜ん」ってな具合。しかし、名前聞いておくんだった。くそ。私はその子と同じ



▲富田 祐樹 東京都
最近、8周年とか100号達成とかもあったんで、「おめでとう」ハガキはいっぱい来ていたんですけど、ほんと、ありがたいことです。



▲木下 義典 佐賀県
あいかわらず大人気のソーサリアンですね。女の人は見たところ魔法使いのようですよ。髪の毛の感じがなかなかいいですよ。

学校に通う3年生である。ふん、一日一善だよ。それだけ！

高橋 政秀(17)東京都
イマイチ押しが甘いなあ。でも、人に親切にするのはいいことです。

◆感熱紙はスパイ用アイテムである。

証明) 必要条件 証拠となる文章が残らない
十分条件 熱、日光であつという間に読めなくなる

証明終わり

びえー！ うっかり日向に置いておいた感熱紙が真っ茶っ茶になっちゃたよー！ LHarcの使い方とか、ROGUEのキー操作とかイロイロプリントしてあったのに。リボン高いし、そのうえいまとっては取り寄せ(CZ-8PC2用なので)だし……。もっと根性のある感熱紙はないのかー！ ドットインパクトプリンタがほしーよー。

松本 康裕(23)広島県

根性がありすぎて字がプリントできないというの困ります。

◆僕はサイボーグ、またはアンドロイドを作りたいと思っています。友人は電子工学科で頭脳を作ってくれるそうです。友人の失敗に備えて、誰か脳を提供してくれませんか？ ボディーができた晩には入れてあげましょう。めざせ！ ダニール・オリバー！ うーん、マッドなハガキだ……。

枝松 樹(20)愛媛県

サイボーグに脳の提供ねえ。死んだあとだったら考えないこともないなあ。いまやると痛いもん。

◆日本国内に生息する蝶は250種くらいいるのですが、この夏、私の標本箱にあらたに1種が加わり、145種が揃いました。ほとんどすべて自分の足で歩き捕虫網をふるって(少々はずかしいんですが)採集し展翅をして標本にしたものです。CARD-PROを使い、現在、日本産蝶類のデータベースを作っています。MacintoshのHyperCardなら電腦図鑑が作れるのになあ……。

斎藤 光一(40)埼玉県

蝶も最近あまり見かけなくなりましたね。さみしいかぎりです。昔は取って食うほどいたんですけどね。

◆数あるパソコン雑誌の中でも内容がいちばんまともでいい。西川善司氏が面白い。余談でよく編集部内の狂っている部分とか書かれているが、そういうメンバーでどうやってちゃんと編集しているのか不思議だ。でも、面白ければなんでもいい。

堤 雅秀(22)神奈川県

信じてくれー。オレは決して狂ってなんかないー。うおー。ガンガンガン。

◆音楽の世界というものは入ってみると非常に面白いものであると思うのです。しかし、楽譜が読めないとか、楽器のひとつもできないとか、作曲なんて人間のできることではないとか、そういったことが頭を過るたびにその世界に入ることためらってしまうことが、まああるのではないのでしょうか。コンピュータミュージックはそういった壁を打破するための、もっとも手取り早い方法たりうとは思いますが。

船山 竜士(21)埼玉県



うーん、想像とは恐ろしい。あのフラッピーがこ
んなふうになるとは。これじゃあ敵のほうを応援
してしまいますね。カワイイもん。



2人とも目がうつろですが、かえってそれで目を
引かせてしまいました。寂しくて鋭いなかを感じ
てしまいました。

大人になってから、「ああ、子供の頃から
なにが楽器を習っていればよかった」とか
思うんですね。楽器やっているとモチるだ
らうし。

◆シムシティ、9つ目のシナリオ。それは…
…、まず空き地がなくなるまで開発する(災害
なしモード)。そして、金をある程度貯めたら
予算カット。災害連発で町を完全に破壊する。
そうすると、一面焼け野原の土地になるので、
そこから町を作る。シナリオ名は戦後の復興。
町を破壊する前の金の貯め具合でレベル設定を
します。一度くらいはやる価値があると思いま
すが、どうでしょう？ 福永 浩司(19)大阪府

それで、町が成長してきたらまた破壊する
ということを繰り返すんですか。人間の歴史
そのものだなあ。
◆某大手電機メーカーのコンピュータH/W技術
者から実家の酒屋にデュードして早半年。最近
は論理回路の飛び回る夢も見なくなってしまい、
とてもさみしいものです。転職前はそれでも日
曜日などに設計でもして……と思っていたが、
実際はそのようなひまもなく、睡眠の補給と
買い物の荷物持ちで終わってしまいます。せ
めて貴誌の記事を見て欲求不満を……なども
考える今日この頃です。林 将智(28)大阪府

日曜日はやっぱり睡眠の補給だけで終わ
りますよね。まあ、休めないときよりはいい
けれど。
◆Oh! X LIVE in '90に初めて掲載されてから、早
2年近くになります。その間いろんなことがあ
りましたが、憧れであった“常連”などとも呼
ばれるまでになりました。私ももうすぐ大学4
年生就職活動もあって忙しくなります。いまで
も十分レポートで大変です。なぜこんなことを
書くかという、そう、もう投稿はしようと思
えて……は、まったくいけません。ミュージック
プログラムはすでに私のライフワークと化して
いるのです。次は“スペシャリスト”の域に達
することができるように気合い入れます。

安藤 正洋(21)青森県

がんばってください。

◆どうもすみません。アンケートハガキの「あ
なたの愛機は」のHD(MB)のところを、僕

はいままでフロッピーのメディアと勘違いし、
2HD(1MB)と書いておりました。ごめんなさい。

柴田 和久(18)東京都

あらあら。

◆Oh! Xを買ってひと通り目を通してから、次号
予告を見て「おお、来月はこういうのがあるの
か」と思って楽しみにしています。しかし、い
ざ買う時には「おお、今月はこういうのをやっ
ているのか」と、すっかり忘れてしまっている
わけなんです。

阪本 泰博(20)大阪府

本屋さんに行く前にちゃんと予告を読んで
予習していきましょう。

◆熱血高校ドッジボール部野球編というのを作
ってほしいなあ。クロスプレイでの乱闘！ 必
殺ピッチャー返し！ バットを砕く魔球！ 絶
対すごいと思う。木全 克徳(21)京都府

なんか、熱血高校というよりは「あばっち
野球軍」になってしまいうぞですね。

◆この間、レンタル電話の申し込みをしたら断
られてしまった。料金はクレジットカードから
落ちるため、まずクレジットカードの審査があ
るのだが、残念ながら私の預金口座は残高がマ
イナス50万円ぐらいであるため、作れなかった
のである。いつになったら通信専用電話が持て
るのだろうか。ちなみにマイナスになったのは
X68000を買ってしまったせいなのだ。一児の
父より。

矢崎 慎一(35)東京都

かわいそうな日本のお父さん。ううっ。

◆電脳倶楽部Tシャツである。コード表が逆さ
まになっているのがいかかわしいが、よお〜
考えてみるとこれはとっても便利なものではな
いか！(特許取ったんですか?)

伊藤 洋司(19)茨城県

あのコード表にはそのほかにも、その人の
体の大きさにあわせて文字の大きさが変わ
るという機能もついているんですよ。

◆ビデオボードの購入を見合わせていたけれど、
10月号の改造記事を見て買い、作ってみました。
別にノイズもなく素人工作にしてはソツなく出
来上がり喜んでます。

伊規須 一男(40)福岡県

日曜大工ならぬ、日曜工作。よかったです
ね。

◆そうだったのか！「OPMというファイルでFM音源演奏可能」というのは、ああやってOPMファイルを作るとのことだったのか。それを知らずに今まで生きてきたのか。ああ、10月号買ってよかった。安部 一馬(23)福岡県
そのまま知らずにいたら一生後悔するところでしたね？

◆うちの学校は生徒の99パーセントが進学希望者である(あくまでも希望)。僕はというと残りの1パーセントの人だったりする。だから、夏休みはバイクや車で(免許取ったのは)遊び回り、バイトで金も稼いだ。おかげさまで、すでにダブる確率50パーセント以上である。同学年の先生で僕の名前を知らない人はいないくらい有名である。……こんな僕はもうどうしたらいいの？ だれか教えてちょうだいな(自分が悪いクセに……)。おっと、バイトの時間だ(自覚のない私)。菅野 弘治(18)東京都

まだ確率が40パーセントくらい残っているじゃないですか。あとは本人次第ですね。

◆テストの前になると、私のノートを借りにくる人がいる。「これで自分と点が同じか、あるいはよかったら割に合わないな」などと、ふと考え込んでしまう。うーむ。

小野寺 光(20)宮城県
貸した相手が単位を取ってるのに、自分は落としたというのがいちばん悲惨なパターンですね。

◆いやあ、盛況でしたね、バックナンバーフェア。午前中はあまり行動しないのですが、この日(9月20日)ばかりは早起きして行ってきました。着いたのは10時30分ごろ。会場の5階はもう若い人でいっぱい(といっても20人ぐらいですが、開店直後のため店内は閑散としていたのでこのコーナーの人だけは普通でなかった)。私も探していた1987年度後半の7冊を無事購入しました。しかし、これは少ないほうでひとりで30冊ぐらいひとかえにしてレジに並ぶ人が多数いたのには驚いてしまいました(重いだろうな)。岡野 英司(42)東京都
そうだったのか。買いにいけばよかった。くそー。

◆この間、「C compiler PRO-68K ver.2.0」を買

いました。噂には聞いていましたが、実際に見てやっぱりすごいなと思いました。あのマニュアル。ちびちび読んでいますが、わからない言葉などがあってなかなかかどりません。なにやらOh!XではC言語の連載が始まったようですが、これぞまさにナイスタイミング。初心者の方にもよくわかるようにお願いします。

石浦 芳仁(20)東京都
ははは。そうではない。あの連載を始めるためにシャープに圧力をかけてver.2.0を発売させたのだ。ウソ。

◆あの、私まだ子供だからよく知らないんですけど、やっぱり大学生になったら機械語という言葉を知りますか？ そ、それで、あ、あの機械語っていうのは、どこの国で話されているのですか。機械と辞典というのはあるのですか？ 加藤 伸(15)千葉県

そんなもんありません。

◆DōGAが大変。CA68は(あ、私のチームです)人手不足、時間不足、資金不足、睡眠不足で大変です。しかも、仕事(グラフィックデザイナー)も忙しい。デザイナーには時間がないのよ。がちょーん。……。でも、がんばってCGAコンテストには出します。安藤 優子(22)福岡県
寝ずにがんばって出してください。

◆難しくてわけのわからないページの下のいろいろ書いてあるのを読んでいた。すると自分の書いたのが載っているではありませんか。編集部の皆様、ありがたき幸せにございます。なんか恥ずかしいものですね。ほかの皆様のと比べると見劣りがします。でも、自慢しようと思うのは人間の性ですね。友人にいいふうそうと思いましたが、Oh!Xの存在すら知らない友達ばかりで……。がんばってください。

笹山 克巳(16)石川県
自慢するついでにOh!Xを宣伝するというのがいいでしょう。

◆SLANG+REDAで開発した作品を手を、某ソフトハウスの入社試験に臨んだところ、なんと内定をいただきました。THE SENTINELのコーナーを支えている方々に感謝しています。いやあ、Oh!Xの読者でよかった。この未熟者の私が合格できたなんてまだ信じられません。

西田 一郎(17)長崎県
いやいや、運がよかったですよ。あつ、違う違う。あなたの実力ですよ。

◆10月号の174頁の溝渕さんへ。JRの比較的古い特急車両のデッキ側の壁(ドアの横)にはちゃんとAC100Vのコンセントが付いています。ただし、プラグが特殊なので変換コードを自作する必要があります。洗面所には電気カミソリ用に普通のプラグの差し込めるコンセントがあるのですが、それを使うとほかのお客さんに迷惑です。津幡 岳弘(23)愛知県

これで、寒い日にコタツを持って電車に乗っても安心ですね。

◆THE USER'S WORKSはいい企画だと思います。ネットをやっていないでもこのようなソフトにお目にかかるのは、普通のユーザーにとっていい刺激になっていいと思います。実際、私もそのひとりです。樋口 雅人(18)福島県
あのコーナーは結構評判がいいんですね。ああいうものって、なにか手作りの味があっていいですからね。

◆私の通っている学校がある町所沢は、ご存じ西武ライオンズの本拠地である。この町のダイエーでは西武優勝に便乗してバーゲンを催すことになっている(優勝したら話だけけれど)。そして、賑やかな商店街のあるほうではなく、駅を挟んで反対側のほうに2軒のパソコンショップが同じビルの中に存在している。ここは秋葉原でもないのに、だ。そこで激しい価格競争が行われるのだが、とうとう2HDのフロッピーディスクのブランド品が一番安いもので680円にもなった。これは秋葉原にも匹敵するものがある。このまま競争が続けば我々消費者はウハウハものである(けど、ここで安定するだろうな)。はたして2店の運命はいかに？

小川 純一(17)埼玉県
あんまりやりすぎて両方ともつぶれたら、みんなさぞかし困るだろうなあ。

◆知り合いにピアノを10年くらいやっているやつがいる。そいつもパソコンユーザーなので、MMLデータの書き方は知っていたのだが、私が「そのデータだけ見てピアノを弾いてみる」というと、妙な具合に顔をそむけて言った。「慣れたらできる。……と思う」。

矢部 尚之(17)大阪府
できたらスゴイ。でも、不可能ではないですよ。

◆ついにプログラマーとして働くことになりました(まだアルバイトの身ですが)。しかも、ゲームソフトのプログラマーです。いままでは市販のゲームにあれやこれやとナンクセをつけていた私ですが、いざ自分で作るとなると、ハードの制約、メモリの制約など目に見えなかったところに問題があることを知り、ゲームに対する見方も「これはどうやっているのだろう？」というふうに変わってきました。パソコンでのプログラミングはあくまで趣味として続けていきたいと思っています。久保 誠(27)京都府
そのへんがむずかしい問題なんですよ。



趣味を仕事にする面白くないという人もいれば、好きなことをやってお金をもらえるなんて最高と思う人もいます。

◆現在、月に十数冊の雑誌を購入している。主に電気、コンピュータ関係の本ばかりである。その他にSFの本も月に数冊買っている。これを1年半続けた結果、ベッドの半分が本箱になってしまった。あと1年たてば、私は本箱の上で寝ることになるかもしれない。

渡部 裕亨(23)福岡県
本を並べてその上に寝ればベッドがいなくなるじゃないですか。変な夢にうなされそうですけど。

◆僕は6畳間に住んでいるんですが、10月号の福岡県の安藤君。4畳半というのは気の毒だが、まだ狭い部屋に住んでいる人も多い。キーボードスタンドや十数台のオーディオ機材、X68000とX1turbo、音源モジュールと雑誌(Oh!X、キーボードマガジン、OPTION、CAR BOY、バリバリマシン等)はまだ我慢する。しかし、どうにもならないっていうのがA1大の製図板だ(マウスマットの代わりにするとGOOD)。友人で下宿にドラフト(?!)があるやつもいるが、全国の機械科諸君! メゲずにがんばろうぜ!

佐藤 仁(21)山梨県
ドラフトですか。ドラム缶のフタじゃ大変でしょう。ちがうって!

◆10月号の特集の電子音楽術入門はいくらなんでもよすぎた。ZMUSIC、FNCなんて最近「あったらいいなあ」と思っていたものがポンと出ていたのだからビックリ。もう、その辺のウン千円もするコンピュータミュージックの解説書のうえをいってます。……とこれだけヨイショしておけば、プレゼントは僕のもの……。あつ、ウソですよ。うん、本当にグッド。

加藤 武史(16)福井県

あの特集は西川善司君が大活躍でしたね。善司先生に励ましのお便りを出そう!?

◆三国志IIで新君主を女性にしてプレイしていると、たまにほかの君主から「わが娘をぜひ貴殿の嫁にさせていただきたい」という密書が来たりするが、君主は女なんだけどもなと思って考えてしまいます。しかし、密書の内容が「わが息子をぜひ貴殿の婿にさせていただきたい」というものだったら、どうだろうか。

藤原 彰人(20)岡山県

でも、プレイしているのは男である君。うーん、なんかややこしい。

◆今年の夏は古くなった扇風機がよく燃えたらしい。うちのはスイッチを入れても最初は動かない。そこでエルボースマッシュを1発食らわすとしばらくして動き出す。だが、うちのやつは燃えたりはしない。なぜなら35年前の扇風機はすべてアルミと鉄でできているから。いった



▲見浦 崇 長野県
耳が尖っている。こいつは人間じゃないな。ローブを着ているほうも怪しい。やい、正体を現わせ。でも、Oh!Xを持っているから許してあげる。

い世の中進歩しているんでしょうかね?

寺尾 文治(38)岡山県

ということは3才のときから使っているわけですね。なんともものもちがいい。でも、押し入れにしようときに重そう。

◆初めて買った。近々、X68000を買う予定なのでOh!Xを買うことにした。本屋でバラバラと見たときは、活字ばかりで非常にかたくるしそうだったが、よく読んでみると非常に面白かった。これからも買っていくつもりだ。

清野 一男(19)秋田県

毎月着々と読者が増える、Oh!X。

ぼくらの掲示板

仲間

- ★X1ディスクユーザーの皆さん! X1を見放すのはまだ早い。このたび、CURECではX1ユーザー総集結号と題して、全国のX1ユーザーを対象にディスクマガジンの制作、発行を予定しています。とにかくX1ユーザーのパワーでなにかすごいことをやろうと考えていますので、ひとりでも多くの方の参加をお願いします。とりあえず下記住所へ62円切手同封のうえご連絡ください。折り返し案内状をお送りします。〒488 愛知県尾張旭市東栄町根の鼻5186-40 水野義則
- ★X68000ユーザーを対象とするサークル「SAKIKO」では会員を募集しています。このサークルではX68000に関する情報提供を中心に活動しています。会員から送られてきた情報を整理してディスクで配布します。誰でも参加できる自由なサークルを目指しているのでよろしく。詳しくは62円切手同封のうえ、下記まで。〒671-12 兵庫県姫路市勝原区山戸241-10 山根邦博(16)

売ります

- ★MZ-2500用のカラースキャナ・ユーティリティ「SS-SC25C」と、ハンディカラースキャナ「WD-05HS」を4万5千円から5万円でお売りします。〒277 千葉県柏市柏715-11 ジェネパレス柏505-101号 加藤康成(19)
- ★オムロンのモデム「MD1200AIII」(1200bps、箱、付属品あり)を9千円で。送料は負担してください。連絡は往復ハガキで。〒755 山口県宇部市上宇部中尾 松尾明法(18)
- ★シャープのプリンタ「CZ-8PG2」を8千円で売ります。6カ月使用、箱、マニュアル、付属品すべてあります。連絡は往復ハガキで。〒562 大阪府箕面市栗生間谷西3-7-9-408 波戸博司(27)

買います

- ★X68000用の1Mバイト以上の増設RAM(CZ-601Cに接続できるもの、完動、付属品つき)を送料込み1万3千円くらいで買います。連絡はハガキで。〒708-15 岡山県久米郡棚原町八神248 磯山直樹(18)

- ★拡張I/Oポート「CZ-8EP」を5千円、コンパクトフロッピーディスク「CZ-3FBD」等(3インチ)を1枚200円、ミニフロッピーディスクドライブ「CZ-80IFS」を1万5千円で。連絡はハガキで。〒福岡県北九州市八幡西区医学生ヶ丘1番5-407号 国藤恭正(38)
- ★MIDIボード「CZ-6BMI」を送料込み1万5千円くらいで。完動品でマニュアル、付属品つきのものを。連絡はハガキで。〒203 東京都東久留米市氷川台2-15-7 佐藤昌(17)
- ★X68000用数値演算プロセッサボード「CZ-6BP1」を3万円以下で(完動、マニュアル、付属品つき)。連絡は往復ハガキで。〒158 東京都世田谷区上用賀4-7-3 伊東雅子(29)
- ★MZ-700/1500用純正ジョイスティック「MZ-1X03」を千円で買います。連絡はハガキで。〒399-07 長野県塩尻市片丘10391 古旗一浩(21)

バックナンバー

- ★Oh!Xの1989年3、4、5月号を送料込み各2千円で。切りぬき以外は可。連絡はハガキで。〒285 千葉県佐倉市城271 伊藤徹(20)

DRIVE ON

●「大人のためのX68000」について。さまざまなデータを扱うときに、住所録というある程度固定されたデータから入るのは正しいと思う。また、エディタなどで住所録を作りだすときにまだ見ぬ明日のことを考えて、項目ごとの区切りを統一させるのは賢明であるが、実際においてはそのときのフォーマット（形式）が非常に難しい。その形式次第であとのそのデータの使われ方はがらりと変わる。その最初の取り決めにはセンスが必要であるが（これはほかのジャンルのソフトにもいえる）、そこを詳しくやっているのは評価される点である。

長谷川 敦士(17) MZ-2500, MSX2 山形県
●今回は辞書の強化に興味がありました。他機種種の辞書を利用するなんて思いもしなかったのですが、思ったより手軽で今度実行して

みたいと思います。X68000にもそろそろ新しいFEPが発売されてもいいころだと思うのですが、10月号を見て「やはりASKも捨てたものじゃない」と改めて思いました。

中川 比呂志(19) X68000, XICs 東京都
●FM音源ですか。世にMIDIが広がるなか、なにか新鮮なものを感じますね。私が初めて組んだプログラムはミュージックプログラムでした。手軽にできるわりに、うまくいったときの感動はとても大きなものですね。ただ、音色の設定には熟練を要するのではないのでしょうか。そこで私がお勧めしたいのは、いまさらながら「SOUND PRO-68K」です。本当に「いまさら」という気はしますが、FM音源を使いこなしたい人にとってはなかなか頼もしいものですし、使えるものです。これで慣らしていけば、しだいに音色設定の達人になれるのではないのでしょうか。15,800円は決して高い買い物ではないように思います。

安井 百合江(16) X68000PRO 愛知県

●「THE USER'S WORKS」について。同人ソフトは安価であり、市販ソフトとはひと味違った「手作りの味」があるので、紹介記事を載せるのは賛成である。制作する側も反響が高ければやりがいがあるはずだ。

泉 昭彦(20) X1turbo, PC-E500 東京都

●「ようこそここへC言語」について。基礎面を中心ということですから、ある程度連載が続いてもレベルを上げないでほしいですね。しかし、それでは記事も行き詰まりを見せてしまうでしょう。だから、まったくの初心者だった人がある程度のプログラムなら作れるというレベルまで上がったと思われるころになったら、大きな（もしくは中くらいの）プログラムを数カ月に分けて完成させるといえるのでしょうか。X-BASIC講座のコラムのときや、(で)のぼーていハンズみたいにです。そういえば昔「ロードランナーで学ぶC言語」とかいう本もありましたね。

高橋 毅(19) X68000PRO, MSX2 埼玉県

ごめんなさいのコーナー

11月号 ごめんなさいのコーナー

ZMUSIC.FNCのところで訂正番地が間違っていました。029Aではなく、0299を2Bにしてください。

7月号 ハードウェア工作入門

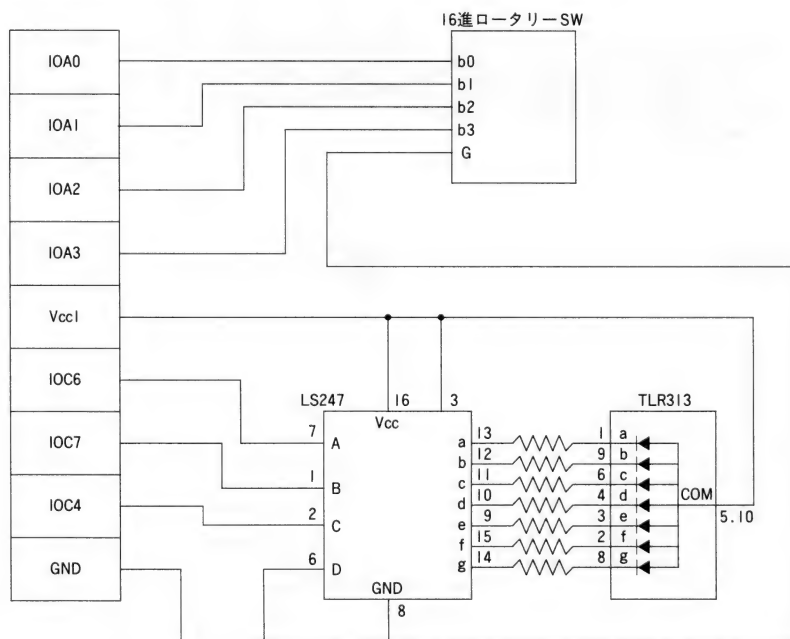
P.107 図2の回路図と8月号に掲載した図1の実体配線図が異なっているようです。これは製作中に変更点があったため、回路図は変更前のものです。図1が正しい回路図です。ご迷惑をかけました。

6月号 INTEGRAL X1

すでにあるファイルと同じファイル名の場合、ファイルサイズが更新されません。下のように訂正してください。

```
0000      1  : KAME-DOS BUG-FIX
0000      2  :
0000      3  :
0000      4  : S-OS REDA
0000      5  :
E042 P    6  #ERR7      EQU #E042
D084 P    7  #FNAM1     EQU #D084
E068 P    8  #FNAM      EQU #E068
0000      9  :
0000     10  :
D565     11  ORG #D565
D565     12  :
D565     13  JP BEGIN
D568     14  OPNSK4
D568     15  :
D568     16  :
DE30     17  ORG #DE30
DE30     18  :
DE30     19  BEGIN
DE30     20  JP NZ, #ERR7
DE33     21  LD HL, #FNAM1+46+22
DE36     22  LD DE, #FNAM+22
DE39     23  LD BC, 13
DE3C     24  LDIR
DE3E     25  JP OPNSK4
DE41     26  :
```

図1 変更後の回路図



バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日 16:00～18:00

お問い合わせは原則として、本誌のバグ情報のみに限らせていただきます。入力法、操作法などはマニュアルをよくお読みください。また、よくアドベンチャーゲームの解答を求めるお電話をいただきますが、本誌ではいっさいお答えできません。ご了承ください。

身近になったC言語 1年ぶりのC特集

▼ようやくC compiler ver. 2.0も発売されました。これを機にC言語を導入された方も多いことでしょう。そしてOh!Xでも1年ぶりのC言語特集です（例えば、昨年の特集もXCの新バージョンにタイミングをあわせて企画していたような気が……）。いまやプログラミング言語の中核となったC言語。ちょっとしたファイル操作から本格的な開発まで、あらゆる分野で活用してください。

▼Oh!Xでは誌面作りを手伝っていただく協力スタッフを募集しています。仕事内容は原稿執筆、プログラム開発、投稿チェックなど。希望者は6000字程度の自由論文を添えて「スタッフ募集」係まで連絡してください。

▼さて、ひそかに予告されていたとおり、来月号はディスク（5"2HD）つきでお届けする予定です。なにが入っているかは開けてみてのお楽しみ。

愛読者プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入の上、希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年12月18日の到着分までとします。当選者の発表は1991年2月号で行います。

10月号プレゼント当選者

①ルーンワース（山形県）築瀬信悦（東京都）倉持聡（徳島県）谷口成広 ②ワールドコート（北海道）谷口有香（神奈川県）角井真吾（大阪府）堀川英知 ③闇の血族（山形県）宮下丈司（東京都）井上綾子（岩手県）泉哲也 ④電脳倶楽部Tシャツ（北海道）佐藤政幸他19名 豆しばり（北海道）白戸知己他9名 えんびつ（栃木県）広田義弘他9名 シャープペン（千葉県）伊藤徹他9名

（敬称略）

以上の方々が当選されました。おめでとうございます。商品に順次発送いたしますが、入荷状況などにより遅れる場合もありますのでご了承ください。

投稿応募要領

- 原稿には、住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は、詳しい内容の説明、利用法、できればフローチャート、変数表、メモリマップ（マシン語の場合）に、参考文献を明記し、プログラムをセーブしたテープ（ディスク）を添えてお送りください。また、掲載にあたっては、編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は、詳しい内容の説明のほかに回路図、部品表、できれば実体配線図も添えてください。編集室で検討のうえ、製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして、他誌との二重投稿、他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル

ソフトバンク出版部

Oh!X「㊟㊟㊟」係

S H I F T ・ B R E A K

▶前期の成績をもらったあと、友達とボーリングを7ゲームをやる。編集室に来てマープルマッドネスをやる。「今日は玉ばっか転がしてるね」と、A.T.氏にいわれる。痛む腰をさすりながら、AMIGAの前に座って必死にトラックボールを転がしている僕。でもみんな、前期の成績まで坂道を転がり落ちるように下がっていたのは内緒だぞ。（純）

▶都内では全部の車を駐車場に入れたとしても10万台以上の車が路上にあふれる計算だそうである。こんな状態で駐車違反を2点にしたところで警察の小遣い稼ぎにしかならない。交通事故が増えると真っ先に責められる警察の立場もちょっとだけ同情するが、交差点内駐車などの迷惑度の高いものから取り締まってほしいものだ。（この前1点減ったH.U.）

▶アサルトコンパネの情報どうも有難うございました。ところで、編集室は「闇の血族」が密かなブーム。もう日常会話中でも「んーもうJESUS」とか「-SIGH」とか「はふ」とか「私はね今、BLUEな気分なんだ」とか手を叩いて「Slap!」とか叫ぶ始末。え？ そんなことしているのは私だけ？ うっそお。（ワタシデナイワタシ善ジデナイ善ジ）

▶コンビニエンスストアでごちゃごちゃと買い物をしたら、金額がちょうど2,000円（税金も合わせて）であった。妙に嬉しい。と思ったら、バイクの走行距離が30,000kmを超える瞬間を見逃してしまった。妙に悔しい。ところで、AMIGA購入計画は資金面で挫折して、増設メモリに転んでしまった。いやあ快適快適、転んでもただでは起きないよ。（A.T.）

▶この号の発売の2日後にはもう、初スキー。気がつけば今年も終わりに近い。そしてこの僕も、X68000を買ってしまった。やはり時は無情に流れていくのだ。そんな状況を反映してか、最近何かと忙しい。でも、こんなときだからこそ自分らしさを失わないようにしたいと思う。最近アウトドアが好きだ。

（今年はシマデスキーのC）

▶ゲームソフトの年末商戦はすごい。特に今年はこれでもかというくらい多くのソフトが発売される。それだけソフトハウスにとっては厳しい戦いとなるわけだ。でも、こういうときって不思議とひとつのゲームが異常に売れちゃったりもするんだよね。あのザナドゥがそうだった。果たして今年は？ ショップへ行行って何が売れるか占ってみよう。（S）

▶なるほど、能書きを垂れないで何もできない連中は多い。能書きを先に垂れては必ず転ぶ。ああ、腐った能書きが多くていけないや。能書きがいけないんじゃない。ああ、精神をナメた肉体と、肉体をナメた精神と、世の中にはどちらかしかないのか？ 心と体の思考のバランス。心の腐ったやつを殴り倒して独裁したい男のロマン、な秋。（K）

▶バージョン2のLK.XはCASH.Xと相性が悪い。XCのバージョン1やGCCからLK.Xを呼び出すと失敗する。そこで、CC.X（バージョン1）とGCC.XをDIS.Xでソースジェネレートし、電脳倶楽部から手に入れたHLK.Xを呼び出すように改造してやった。これで世界に平和が戻った。それにしてもバージョン2のコンパイル速度は遅すぎるとは思いません？（K0）

▶この間までは「部屋にクーラーがない」とかいて騒いでいたのですが、涼しくなってきたと周りを見わたすと暖房器具がなにもない。しいて挙げるなら、布団と2台のコンピュータぐらいでしょうか。4月に上京してきたところだとはいえ、なんとも情けないかぎり。コタツでも買うしかないかな。ちょっと、机もないことだし。（A）

▶最近体力がひどく落ちてきたので「このままでは死んじゃうかも……」と思い、ジョギング用にスニーカーを買ってきた。が、それも「徹夜明けの身体で走ったら心臓マヒで死んじゃうかも……」という心配に代わっただけ。結局そのスニーカーを履いて、家でダンスエクササイズとバーレッスンをやっている今日この頃の私……情けない……。 （E.O.）

▶「贅沢を持つ喜び」も捨てがたいが、とりあえずA500を買った。これで3台目、あともう1台は誰かに……、あ、MIDIも買わなきゃ。さて、来月号は特別定価780円だ。そうそう、生ディスクの用意も忘れないように。そんでもって、SX-WINDOWは本当にやってくるぞ。RAMの準備はいいか？ ハードディスクに余裕はあるか？ （U）

▶かつては70万円もしたMacintosh SEの相当品が19万8千円。当時みんなの憧れだったMac IIに相当するLC（68020、2MB、256色カラー）もEXPERTより安い。まあ冷静に見積ればそんなものかなという気もするが、X68000のハードは重装備だからそれほど安くはならないだろうな。うーむ。（T）

microOdyssey

私の机にはSS-NETの電話機が載っている。シャープ製だが、子機はない。そのかわり、こいつには立派なRS-232C端子があって、同じ機のX68000とつなげばUNIXの端末になる。つまりSS-NETというのは社内の電話回線を使ってLANを構築できるシステムである。凄いやつだ。目のつけどころがシャープである。

さて、編集部電話機は交換機が変わったり、引っ越したりで、何回も変わっている。そのたびに、電話機の操作に戸惑い、外線を誤って切り、内線番号表を書き換え、さらには100件近い短縮番号を登録してきたわけだ。

そこで今回は電話の悪口を書く。へんに思想めいたことより、はっきり悪口と決めたほうが書きやすい。まあ電話料金が安いとかいったことは今回は見逃してあげようと思う。

実は、そのSS-NETの電話機にもいささかの不満がある。だって、今どき液晶表示がないんだもの。以前使っていた電話機では、かけた番号を液晶パネルにエコーバックして確認することができた。簡単なことだが、あるとないでは大違い。リダイヤルや短縮番号を使う際にも、相手の番号が表示されるのは便利である。表示がないと、短縮番号を登録しても、実際に相手に電話をかけてみないことにはちゃんと登録されているかさわからない。これって間抜けな話でしょ。

ところで、東京都では局番が4桁になる。市内通話に8桁の番号が必要なのだ。覚えるのも大変だが、番号を入力ミスする確率も確実にアップする。緑の公衆電話にはカードの残り度数を大きな文字でデジタル表示する。それができるんだしたら、エコーバックしてくれよ。と言いたい。後ろの人に相手の番号を見られて困るなら、表示をOFFにするくらい簡単でしょ。うう、きっと考えてないんだな。

じゃあ、表示のある電話機ならいいかというところでない。表示があっても電話番号はエディットできない。市外番号だと10桁にも及ぶのに、最後の1桁を間違えただけで最初からやり直しだ。

ここでちょっと考えてほしい。東京都の電話番号が足りないのはわかる。でもいまある3桁の局番の頭に3をつけて4桁にするというのはなんかおかしい。新しい局番だけ4桁にすればいいのでは？ ところがこれがダメ。電話にはエンドコードというものがいないからだ。たとえば、1234-5678という局番4桁の番号作ったとしても、これを局番3桁の123-4567という番号と区別するできないのである。

もう桁数についてはしょうがない。あとは今後設置される電話機に便利になってもらうしかない。私としては、ある程度の大きさの液晶表示とメモリをつんで電話番号を管理してほしい。登録した番号は別の電話機に転送できるようにしてほしい。メモリカードを公衆電話に使えるようにしてほしい。そうそう、プッシュボタンの配列は電卓と同じにほしい。とにかくなんとかしてほしいのだ。

その昔、プッシュホンに変えると計算機能があると聞いて感動したが、それは電話機に電卓機能があるのではなく、電話をかけて計算してもらうサービスのことだった。今はそういう時代じゃない。(T)

1991年1月号12月18日(火)発売

特集1 急接近! SX-WINDOWのすべて

特集2 謹賀新年PRO-68Kの謎を探る

Oh! X 3周年記念特大プレゼント第2弾

特別付録5"2HDディスク

特別定価780円(消費税込)

バックナンバー常備店

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(233)3312	神奈川	厚木	有隣堂厚木店 0462(23)4111
	//	書泉ブックマートB1 03(294)0011		平塚	文教堂四の宮店 0463(54)2880
	//	書泉グランデ5F 03(295)0011	千葉	柏	新星堂カルチェ 5 0471(64)8551
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660		船橋	リプロ船橋店 0474(25)0111
	八重洲	八重洲ブックセンター3F 03(281)1811	//	//	芳林堂書店津田沼店 0474(78)3737
	新宿	紀伊国屋書店本店 03(354)0131	千葉	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
	高田馬場	未来堂書店 03(200)9185	埼玉	川越	黒田書店 0492(25)3138
	渋谷	大盛堂書店 03(463)0511		川口	岩瀬書店 0482(52)2190
	池袋	リプロ池袋店 03(981)0111	茨城	水戸	川又書店駅前店 0292(31)0102
	//	西武百貨店9F コンピュータ・フォーラム 03(981)0111	大阪	北区	旭屋書店本店 06(313)1191
	神奈川	横浜	都島区	都島区	駿々堂京橋店 06(353)2413
	//	有隣堂横浜駅西口店 045(311)6265	京都	中京区	オーム社書店 075(221)0280
	藤沢	有隣堂藤沢店 0466(26)1411	愛知	名古屋	三省堂名古屋店 052(562)0077
			//	//	パソコンΣ上津津店 052(251)8334
			刈谷	刈谷	三洋堂書店刈谷店 0566(24)1134
			長野	飯田	平安堂飯田店 0265(24)4545
			北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ、必要事項を明記のうえ、郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので、大切に保管してください。なお、すでに定期購読をご利用の方には期限終了の

少し前にご通知いたします。継続希望の方は、上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店、日本IPS(株)にお申し込みください。なお、購読料金は郵送方法、地域によって異なりますので、下記宛必ずお問い合わせください。

日本IPS株式会社

〒101 東京都千代田区飯田橋3-11-6

☎03(238)0700



12月号

■1990年12月1日発行 定価560円(本体544円)

■発行人 孫正義

■編集人 橋本五郎

■発売元 ソフトバンク株式会社

■出版事業部 〒108 東京都港区高輪2-19-13 NS高輪ビル

Oh!X編集部 ☎03(5488)1309

出版営業部 ☎03(5488)1360 FAX 03(5488)1364

広告センター ☎03(297)0181

■印刷 凸版印刷株式会社

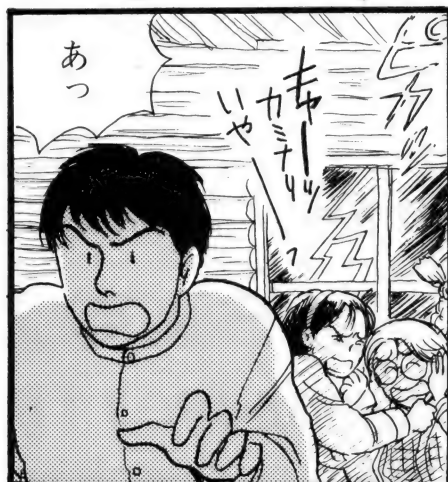
©1990 SOFTBANK CORP. 雑誌 02179-12本誌からの無断転載を禁じます。

落丁・乱丁の場合はお取り替えいたします。



満開の電子ちゃん

作: いわい いっぺい
え: 岡村 繁



だ、だ、電気は?

購読方法: 通信販売でのみ扱っております。御注文は、現金書留または郵便振替で、定期購読料6ヶ月分6,000円(送料サービス、消費税込)を下記の宛先へお送り下さい。

●現金書留の場合:

〒171 東京都豊島区要町1-19-3 いさみビル4F 満開製作所

●郵便振替の場合:

東京 5-362847 満開製作所

※御注文の際は、郵便番号・住所・氏名・電話番号を忘れずに御記入下さい。

●お問い合わせ先 TEL(03)554-9282(月~金 午前11時~午後6時)

●11月18日以降に受け付けた分は、原則としてVol.31から発送します。

新たに購読を希望される方は、「新規」と御明記下さい。

(製品の性格上、返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします)

(バックナンバーの受付は、購読者の方に限らせていただきます)

「電脳倶楽部の購読を始めたから成績はアップ、部活ではレギュラー、おまけに彼女までできました」という文章を何年も使い回すほど年季がし、今までにここからデビューした傑作の数々には目を見はるものがあります。CGに革命を起こした「PIC」や、ファイル管理ツール「F」等を手放すのは、酒をやめるより苦しいに違いありません。これらは当然、品切れなしのバックナンバーでいつでも入手できます。デバッグやバージョニングのアフターケア(再掲載)も嬉しいですね。



平木敬太郎
(福井県)

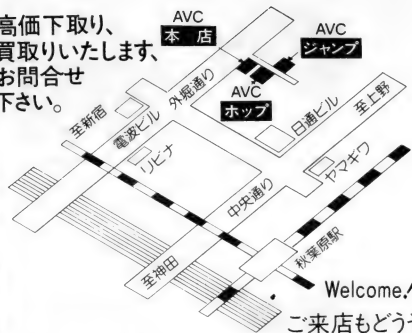


AVCフタバ

03(253)7661

〒101 東京都千代田区外神田3-2-3 ☎03-253-7611(代)

高価下取り、
買取りいたします、
お問合せ
下さい。



Welcome!
ご来店どうぞ。

今すぐ もよりの電話から	仙 台 022-264-3704	名 屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

△68000 待望の新しい仲間登場!!

PERSONAL WORKSTATION
EXPERT II・EXPERT II HD



EXPERT II・EXPERT II HD
集積度を高めた「マンハッタンシェイプ」3Mの大容量メモリを搭載。本格的なウィンドウシステム、SX-WINDOW搭載。

(写真のモニタは別売です。)

CZ-603C 標準価格¥338,000
CZ-613C 標準価格¥448,000

AVC特価

△68000

PERSONAL WORKSTATION
PRO II・PRO II HD



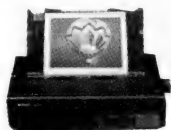
PRO II・PRO II HD
拡張I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。もちろん、SX-WINDOW搭載。

(写真のモニタは別売です。)

CZ-653C 標準価格¥285,000
CZ-663C 標準価格¥395,000

AVC特価

CZ-8PC4



48ドット熱転写プリンター。精密な文字、ハードコピーも可能。

CZ-8PC4.....¥ 99,800

AVC特価¥ ???

お勧めディスプレイコーナー 組合せは自由、価格はお気軽にご相談下さい。

CZ-604D
標準価格¥94,800
AVC特価

- 0.31mmドットピッチ
- 2モードオートスキャン
- ステレオスピーカー搭載
- チルト台同梱

CU-21HD
標準価格¥148,000
AVC特価

- 0.52mmドットピッチ
- 21型ディスプレイ
- 3モードオートスキャン
- ステレオスピーカー搭載

CZ-613D
標準価格¥135,000
AVC特価

- ドットピッチ0.31mm
- TVチューナー搭載
- ステレオスピーカー搭載
- チルト台同梱

CZ-605D
標準価格¥115,000
AVC特価

- ドットピッチ0.39mm
- TVチューナー搭載
- ステレオスピーカー搭載
- チルト台同梱

CZ-603D
標準価格¥84,800
AVC特価

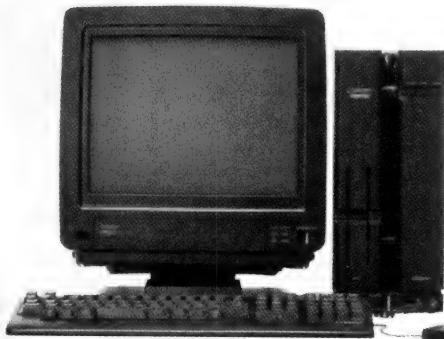
- 0.31mmドットピッチ
- TVチューナー無し
- 3モードオートスキャン
- チルト台同梱

CZ-602D
標準価格¥99,800
AVC特価

- ドットピッチ0.39mm
- TVチューナー搭載
- チルト台同梱

△68000

PERSONAL WORKSTATION
EXPERT HD



CZ-612C-BK¥466,000
CZ-602D-BK¥ 99,800

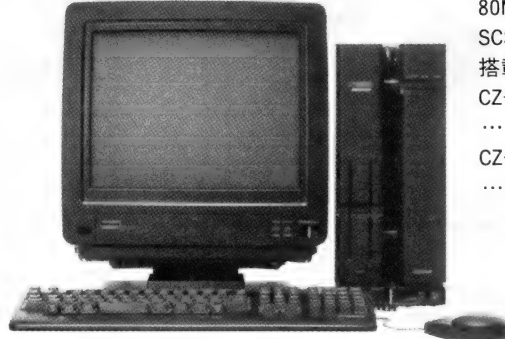
セットでお買上の方に、
SX-WINDOWジョイカード、
「グラデウス」ディスク
セット10枚プレゼント!

AVC特価

¥368,000

△68000

PERSONAL WORKSTATION
SUPER HD



80MBハードディスク、
SCSIインターフェース
搭載!
CZ-623C-TN¥498,000
CZ-613D-TN¥135,000

AVC特価

お電話で.....

●頭金なし(手軽な電話クレジット) ●製品先取り(お支払いは約1~2ヶ月後から) ●低金利クレジット(1回の支払いは2,700円以上で3~48回。ボーナス併用可) ●カレッジクレジット(保証人なし。但し満20歳以上の学生の方) ●18歳未満の方(ご両親が代理購入者としてお申し込み下さい) ●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい) ●完全保証(すべてメーカー保証書付。アフターケア万全) ●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時
まで受付 日曜・祝日も営業

●セットの組合せは自由、広告に出ていない他の機種はお問合せ下さい。

株式
会社

デンキヤ



営業時間AM11:00~PM7:00 水・木曜定休

セット超特価

68000

PERSONAL WORKSTATION

PRO II・PRO II HD

CZ-653C

CZ-604D

セット¥特価

¥24,400×12回

¥13,300×24回

CZ-653C

CZ-605D

セット¥特価

¥25,700×12回

¥13,700×24回

CZ-603C

CZ-604D

セット¥特価

¥27,500×12回

¥14,600×24回

CZ-603C

CZ-605D

セット¥特価

¥28,800×12回

¥15,300×24回

(価格は全て税込みです)

セット超特価

68000

PERSONAL WORKSTATION

EXPERT II・EXPERT II HD

CZ-663C

CZ-605D

セット¥特価

¥32,800×12回

¥17,400×24回

CZ-663C

CZ-613D

セット¥特価

¥34,000×12回

¥18,100×24回

CZ-613C

CZ-613D

セット¥特価

¥36,900×12回

¥19,600×24回

CZ-623C

CZ-613D

セット¥特価

¥40,600×12回

¥21,600×24回

全品メーカー保証 即決クレジットOK

ディスプレイ

CZ-604D

特価

CZ-605D

特価

CZ-613D

特価

CU-21HD

特価

プリンタ

CZ-8PC4

特価

CZ-8PG1

特価

CZ-8PG2

特価

AP-850

¥58,000

周辺機器

CZ-8NJ1

¥1,400

CZ-8NJ2

¥18,540

PIO-6BE1A

¥20,000

PIO-6BE2

¥39,000

ソフト

CZ-213MS

¥15,500

CZ-259SS

¥5,200

CZ-219SS

¥23,100

CZ-245LS

¥35,500

24時間テレホンサービス

0482-54-3444

お申し込み

TEL.0482-54-3400

FAX.0482-54-3443

埼玉県川口市西川口4-6-4

お支払い

下記取引銀行口座

までお振込み下さい。

三菱銀行西川口支店

(株)デンキヤ 0258081

SHARP X68000

Apple Computer

Software Hardware **DEVELOPMENT**

EWS SUN Next

21世紀のテクノロジーリードする *First Class Technology*

Print jack

KGU-180A

¥38,000

GiGa TRASH

1GByteDAT Drive
FCT-120G

¥498,000(予価)

光磁気ディスクドライブ

Mo TRASH

FCT-060M

¥498,000(予価)

First Class Technology

大容量ハードディスク
200MB外部高速ドライブ

Mac Box II

FHD200

¥298,000(予価)

計測制御ユニット

SCSI対応

Mac Pio II

FCT-16/16PIO

¥298,000(予価)

カラー立体画像入力装置

Hi SCAN

FCT-600SS

¥458,000(予価)

計測制御ユニット

SCSI対応

Mac Adc II

FCT-16/12ADC

¥358,000(予価)

開発スタッフ募集中

社会保険完備・海外長期出張研修制度
Macintosh UNIXが何よりも好きな若い力求む
サンフランシスコに開発室分室あり

販売代理店募集中

開発元 First Class Technology

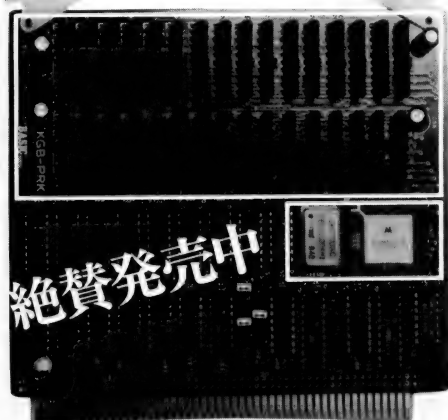
11-18 Kyo-machi Utsunomiya-shi Tochigi-ken Japan 320
TEL 0286-38-0301 Fax 0286-38-0305

販売元 Keisoku Giken Corp.

503-1 Takebayashi-machi Utsunomiya-shi Tochigi-ken Japan 321
TEL 0286-22-9811 Fax 0286-25-3970

2枚のボードが1枚になった

KGB-X68PRK



※写真はKGB-X68PRK-14です

※拡張I/O BOXでは動作しません。

※CZ-600C、601C、611C、652C、653C、662C、663Cで御使用の際にはあらかじめ専用の1Mメモリ(CZ-6BE1、A、B等)でメインメモリを2Mバイト以上にしておく必要があります。

広大なメモリ空間を実現する最大4Mバイトの

高速増設メモリ

高速演算を約束してくれる

数値演算プロセッサ

- メモリアクセスノーズウェイトによる高速アクセス
- CZ-6BE2、CZ-6BE4、CZ-6BP1との混在が可能
- 複数枚のKGB-X68PRKの実装が可能
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能
- ジャンパの変更により数値演算プロセッサの1枚目、2枚目、未使用の選択が可能
- 1M、2M、3Mメモリモデルは購入後もメモリ増設が可能
- PRK-10、11、12、13、14にはデバイスドライバ(FLOAT3.X)が付属

製品価格一覧

KGB-X68PRK-00	¥34,000
(メモリ無し、数値演算プロセッサ無し)	
KGB-X68PRK-01	¥58,000
(1Mメモリ 数値演算プロセッサ無し)	
KGB-X68PRK-02	¥74,000
(2Mメモリ 数値演算プロセッサ無し)	
KGB-X68PRK-03	¥98,000
(3Mメモリ 数値演算プロセッサ無し)	
KGB-X68PRK-04	¥122,000
(4Mメモリ 数値演算プロセッサ無し)	
KGB-X68PRK-11	¥96,000
(1Mメモリ 数値演算プロセッサ付き)	
KGB-X68PRK-12	¥112,000
(2Mメモリ 数値演算プロセッサ付き)	
KGB-X68PRK-13	¥136,000
(3Mメモリ 数値演算プロセッサ付き)	
KGB-X68PRK-14	¥160,000
(4Mメモリ 数値演算プロセッサ付き)	

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ	
MC68881RC16	¥38,000

PRK質問箱

Q、購入後のメモリ増設はどうやるのでしょうか？

A、ご購入後のPRKに対するメモリの増設は半田付け等の技術を要するため原則として当社に送り返していただき増設いたします。自分でメモリ増設をする場合は通信販売のみですが必要な部品の販売も致します。御希望の方はお問い合わせ下さい。

Q、数値演算プロセッサにMC68882を使用することは可能ですか？

A、MC68882では動作しないソフトが存在するため使用できません。

Q、「数値演算プロセッサのみ」や「プロセッサ無しメモリ無し」のPRKがほしいのですが？

A、PRK-10、PRK-00の型番で商品化しております。

※最近PRKをスロットに挿入したが動作しないと言う御質問を良く受けますが、ほとんどの場合は差し込み不足が原因です。X68000のスロットは大変堅く裏蓋が閉まる状態で差し込み不十分場合があります。御注意ください。

充実のBASIC HOUSEソフトウェア&ハードウェア

高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥42,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥68,000
ハンディプリンタ & インターフェース(HANDYPRINTjack) X68000	¥24,800
高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥98,000
汎用ローコストA/D & PIOボード(KGB-X1S) X1	¥19,800
高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
64180CPUボードMach 180(KGB-CPXB) X68000	¥98,000
ローコストMIDIインターフェース(MELODY BOX) X68000	¥16,800

BASIC拡張関数パッケージ(B6-6301)	¥9,800	C言語ライブラリ(B6-6305)	¥6,800
ディスクキャッシュ(B6-6304)	¥6,800	Toys & Tools (B6-6307)	¥6,800
BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	¥14,800		
アイコンエディタ(B6-6303)	¥4,800	CP/M68Kエミュレータ(B6-6302)	¥19,800

おしらせ

DISK CACHER Version UP

皆様に御愛用いただいているディスクキャッシャーが高速化(従来比平均3倍)を行ないVer. UPいたしました。今回のVer. UPはハードディスクキャッシャーのみでHD-DISKCACHE Ver 2.0未満のキャッシャーを御持ちの方がVer. UPの対象となります。御希望の方は旧バージョンのディスクのラベルと代金¥1,500(送料、税込み)を同封して現金書留で御申し込み下さい。

ビデオボードを外付けに!! ビデオボードケース(KGB-BVBX)

通販のみ発売中

SHARPより発売されているCZ-6BVIを外付けにする、ケースです。このケースの使用によりあなたのX68000のスロットが開放されます。

Human68k下のソフトのCRT出力を強制的に15k HZ出力にする(768×512モード除く)
おまけユーティリティ付き

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部
大田原営業所/マイコンショップ

宇都宮市竹林町503-1 TEL0286-22-9811 FAX0286-25-3970
大田原市美原1-13-4 TEL0287-23-5352 FAX0286-23-5364

マイコンショップ

BASIC HOUSE

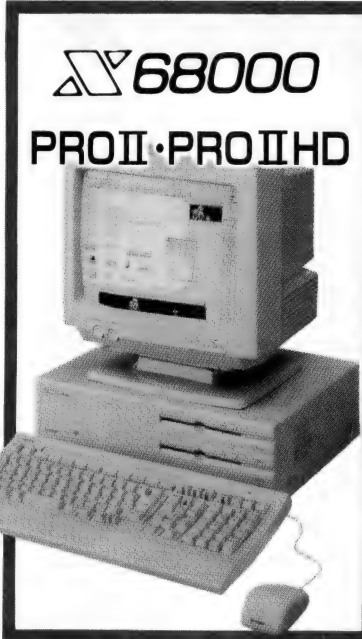
お申し込み・お問い合わせは

0286-22-9811(代)

下取り・買取り大観迎!!

下取り差額のお支払いは、クレジットを御利用下さい。

TV 日本テレビ、TBS、フジテレビ、
テレビ朝日、テレビ東京系列
でCM放映中//直営10店舗



基本セット PROII-HD **月々¥2,300から**

クレジット注文 NO.10

CZ-663C (本体) ¥Sofmap特価
CZ-605D (15"ドットピッチ0.39) ¥Sofmap特価
マクセルブランクディスク 5"2HD×10枚 ¥Sofmap特価
標準価格 ¥511,000 ¥お電話にて

¥2,300×72回	ボーナス ¥30,000×12回
¥4,900×60回	ボーナス ¥20,000×10回
¥8,300×60回	ボーナス なし
¥12,600×36回	ボーナス なし
¥18,100×24回	ボーナス なし

ビジネスセット PROII-HD **月々¥2,300から**

クレジット注文 NO.11

CZ-663C (本体) ¥Sofmap特価
CZ-605D (15"ドットピッチ0.39) ¥Sofmap特価
CZ-8PG2 (24ピン漢字ドットプリンター-130dpi) ¥Sofmap特価
CZ-212BS (CARD PRO-68K) ¥Sofmap特価
マクセルブランクディスク 5"2HD×10枚 ¥Sofmap特価
標準価格 ¥739,000 ¥お電話にて

¥2,300×72回	ボーナス ¥50,000×12回
¥6,100×48回	ボーナス ¥50,000×8回
¥9,700×84回	ボーナス なし
¥12,000×60回	ボーナス なし
¥14,400×48回	ボーナス なし

データベースセット PROII-HD **月々¥2,300から**

クレジット注文 NO.12

CZ-663C (本体) ¥Sofmap特価
CZ-613D (15"ドットピッチ0.31) ¥Sofmap特価
CZ-8PG1 (24ピン漢字ドットプリンター-80dpi) ¥Sofmap特価
CZ-226BS (CARD PRO-68K) ¥Sofmap特価
CZ-220BS (DATA PRO-68K) ¥Sofmap特価
マクセルブランクディスク 5"2HD×10枚 ¥Sofmap特価
標準価格 ¥748,800 ¥お電話にて

¥2,300×84回	ボーナス ¥45,000×14回
¥5,500×60回	ボーナス ¥40,000×10回
¥9,800×84回	ボーナス なし
¥13,500×54回	ボーナス なし
¥18,600×36回	ボーナス なし

基本セット PROII **月々¥3,200から**

クレジット注文 NO.13

CZ-653C (本体) ¥Sofmap特価
CZ-605D (15"ドットピッチ0.39) ¥Sofmap特価
マクセルブランクディスク 5"2HD×10枚 ¥Sofmap特価
標準価格 ¥401,000 ¥お電話にて

¥3,200×60回	ボーナス ¥20,000×10回
¥5,300×48回	ボーナス なし
¥7,800×48回	ボーナス なし
¥10,000×36回	ボーナス なし
¥14,400×24回	ボーナス なし

通信セット PROII **月々¥1,800から**

クレジット注文 NO.14

CZ-653C (本体) ¥Sofmap特価
CZ-613D (15"ドットピッチ0.31) ¥Sofmap特価
CZ-8PG1 (24ピン漢字ドットプリンター-80dpi) ¥Sofmap特価
MD-24FS5 (通信モデム2400BPS) ¥Sofmap特価
CZ-257CS (Communication PRO-68K Ver.2) ¥Sofmap特価
マクセルブランクディスク 5"2HD×10枚 ¥Sofmap特価
標準価格 ¥620,600 ¥お電話にて

¥1,800×60回	ボーナス ¥50,000×10回
¥5,400×36回	ボーナス ¥60,000×6回
¥9,000×72回	ボーナス なし
¥12,100×48回	ボーナス なし
¥15,400×36回	ボーナス なし

プリントセット PROII **月々¥2,200から**

クレジット注文 NO.15

CZ-653C (本体) ¥Sofmap特価
CZ-605D (15"ドットピッチ0.39) ¥Sofmap特価
CZ-8PC4 (48ピン熱転写プリンター) ¥Sofmap特価
CZ-221HS (NEW Printshop PRO-68K) ¥Sofmap特価
CZ-235GS (グラフィックライブラリVOL.1) ¥Sofmap特価
CZ-236GS (グラフィックライブラリVOL.2) ¥Sofmap特価
マクセルブランクディスク 5"2HD×10枚 ¥Sofmap特価
標準価格 ¥538,200 ¥お電話にて

¥2,200×60回	ボーナス ¥40,000×10回
¥4,700×48回	ボーナス ¥35,000×8回
¥7,100×84回	ボーナス なし
¥9,800×54回	ボーナス なし
¥13,500×36回	ボーナス なし

どこよりもお得
合しかない!
差額クレジット
OK!!

下取り差額は
随時変動します。 **高額下取り差額表**
その他の商品についてもお電話でお気軽にお問い合わせ下さい。

お送りになる方、又は
直接大阪店に来られる方 0120-110-833
商品の先取り先 千代田区外神田1-3 野木ビル3F
ソフマップ買取センター 08-641-8801

高額買取価格表

あなたが今、欲しい機種 (新品)

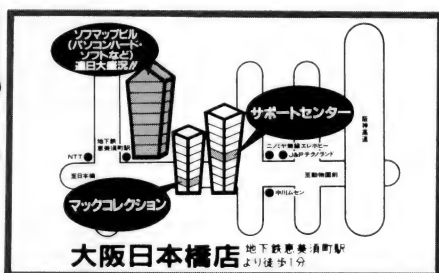
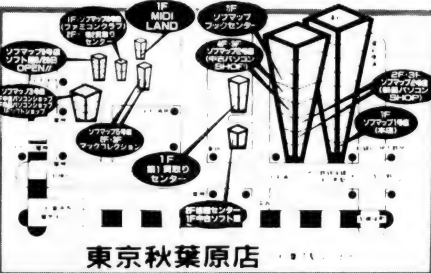
	SUPER-HD CZ-623C CZ-613D	EXPERT II CZ-603C CZ-605D	EXPERT II HD CZ-613C CZ-613D	PRO II CZ-653C CZ-605D	PRO II-HD CZ-603C CZ-605D
下取り機種	交換差額	交換差額	交換差額	交換差額	交換差額
CZ-652C	¥308,000	¥135,000	¥238,000	¥95,000	¥135,000
CZ-602D	¥268,000	¥95,000	¥198,000	¥55,000	¥95,000
CZ-602C					
CZ-602D					
CZ-611C	¥270,000	¥97,000	¥200,000	¥57,000	¥97,000
CZ-611D					
CZ-601C	¥318,000	¥145,000	¥248,000	¥105,000	¥145,000
CZ-601D					
CZ-600C	¥323,000	¥150,000	¥253,000	¥110,000	¥150,000
CZ-601D					
CZ-880C	¥440,000	¥267,000	¥370,000	¥227,000	¥267,000
CZ-880D					
PC-9801VX21	¥323,000	¥150,000	¥253,000	¥110,000	¥150,000
PC-KD854N					
FM-TOWNS-2	¥398,000	¥225,000	¥328,000	¥185,000	¥225,000
FMT-DP531					

商品名

高額買取

X68000 モニターセット

X68 (CZ-662C + CZ-600D/601D)	¥250,000
X68 (CZ-662C + CZ-611D/612D)	¥260,000
X68 (CZ-652C + CZ-600D/601D)	¥210,000
X68 (CZ-652C + CZ-611D/612D)	¥220,000
X68 (CZ-623C + CZ-602D)	¥360,000
X68 (CZ-623C + CZ-605D)	¥380,000
X68 (CZ-623C + CZ-613D)	¥390,000
X68 (CZ-623C + CZ-603D)	¥345,000
X68 (CZ-623C + CZ-604D)	¥350,000
X68 (CZ-612C + CZ-600D/601D)	¥290,000
X68 (CZ-612C + CZ-611D/612D)	¥300,000
X68 (CZ-611C + CZ-600D/601D)	¥235,000
X68 (CZ-611C + CZ-611D/612D)	¥245,000
X68 (CZ-603C + CZ-602D)	¥255,000
X68 (CZ-603C + CZ-605D)	¥270,000
X68 (CZ-603C + CZ-613D)	¥280,000
X68 (CZ-603C + CZ-603D)	¥215,000
X68 (CZ-603C + CZ-604D)	¥225,000
X68 (CZ-602C + CZ-600D/601D)	¥240,000
X68 (CZ-602C + CZ-611D/612D)	¥250,000
X68 (CZ-601C + CZ-600D/601D)	¥195,000
X68 (CZ-600C + CZ-600D/601D)	¥190,000



業界No.1の低金利

支 払 回 数	1	3	6	10	12	15	18	20	24	30	36	42	48	54	60	66	72	78	84
他 社 金 利	3	4	5	7	9	10	12	13	16	19	21	25	28	31	35	—	—	—	—
Sofmap金利	2.0	3.0	4.0	5.5	5.5	8.5	11.0	11.0	11.5	16.0	16.0	20	21	26	27	33	35	39	42

お支払い方法

- 代金引換システム**
係員が品物をお届けに行きますので、その時にお支払い下さい
- クレジット**
お電話で支払い回数、支払い開始日、ボーナスの有無をおつや下下さい
こちらからクレジット用紙をお送り致しますので、ご記入・ご捺印の上ご返送下さい
商品到着後、御指定の口座から自動引落しとなります
- 銀行振込**
お電話で御注文の下、下記振込先へ電信扱いでお振り込み下さい
ご確認後、たたちに商品をお送りします 振込手数料はお客様負担となります

店頭で直接来られる方は **【東京03-258-3156】【大阪06-647-0562】**

通信販売を
ご利用の方は **☎ 0120-110-833 【大阪06-633-7224】**

FAX.03-253-4280

札幌 011-885-7030
仙台 022-268-3405
新潟 0252-22-6139

横浜 045-311-3441
金沢 0762-21-7045
名古屋 052-332-2117

広島 082-222-0604
福岡 092-752-0044
高松 0878-34-8833

24時間テレホンサービス ☎ **03-258-7910**

フリーダイヤル 商品発送のお問合わせ ☎ **0120-08-0113**

フリーダイヤル 故障・修理のお問合わせ ☎ **0120-11-0292**

振込先 東京秋葉原店
三和銀行秋葉原支店(普)1012131
口座名義 株ソフマップ

電話受付時間/年中●平 日 AM11:00~PM8:00
●無休 ●日・祭日 AM10:00~PM7:00

株式会社 **ソフマップ**
旧名 マップジャパン

〒101 東京都千代田区外神田3丁目15番6号小暮末広ビル1F
〒556 大阪府浪速区日本橋5丁目7番17号ソフマップビル

●掲載の商品以外にも多数取り扱っておりますので、お気軽にお問い合わせ下さい。又、商品在庫は毎日変動しますので、品切れの際は御予約承ります。

ALBIT

アイビット電子株式会社

SHARP

パソコン本体から周辺機器まで品数取り揃え 大特価セール実施中!!

型名	品名	正価	特価	型名	品名	正価	特価	型名	品名	正価	特価
UE-1D02	15インチカラーディスプレイAXシリーズ	158,000	特価	CZ-6BE2 2M	X68000増設RAMボード	79,800	特価	MZ-1R11	MZ-5500 256KRAM	80,000	35,000
UE-1D03	15インチカラーディスプレイAXシリーズ	123,000	特価	CZ-6EB1	拡張 I/O BOX	88,000	69,800	MZ-1R36	MZ-28611M 増設RAM	45,000	15,000
UE-1E03	5FDインターフェイスカード AXシリーズ	28,000	特価	CZ-6ST1	チルトスタンド	5,800	3,500	MZ-1R35	MZ-28611M 増設RAM	55,000	19,000
UE-1E02	AX286LICカードI AXシリーズ	45,000	特価	CZ-6SD1	システムラック	44,800	38,000	MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
UE-1E04	S-RNインターフェイスカード AXシリーズ	70,000	特価	CZ-6TU G.B	RGBシステムチューナー	33,100	26,500	MZ-1R16	MZ-5500 128KRAM	30,000	8,000
UE-1P01	136桁漢字プリンタ AXシリーズ	268,000	特価	CZ-822C	X1G MODEL 30	118,000	35,000	MZ-1R27A	MZ-2500 V RAM	13,000	10,000
UE-1P02	速136桁漢字プリンタ AXシリーズ	550,000	特価	CZ-820C	X1G MODEL 10	69,800	16,800	MZ-1R26	MZ-2500 増設RAM	15,000	12,800
UE-1P05	6桁漢字水平プリンタ AXシリーズ		特価	CZ-888C	XI TURBO X3	169,800	95,000	MZ-1R21	漢字ROM	38,000	13,000
UE-1P04	136桁漢字プリンタ AXシリーズ		特価	CZ-8BGR2	グラフィックボード X1	14,800	3,000	MZ-1R24	MZ-1500 辞書ROM	22,000	6,000
UE-1P03	80桁漢字プリンタ AXシリーズ		特価	CZ-8BF1	FDインターフェイス	14,800	11,500	MZ-1R32	MZ6500RAM	80,000	40,000
UE-1R04	2M RAMボード AXシリーズ	180,000	特価	CZ-8BK2	漢字ROM	19,800	16,800	MZ-1R31	漢字ROM	28,000	20,000
UE-1R03	2M RAMボード AXシリーズ	100,000	特価	CZ-8BM2	232C マウスセット	19,800	16,800	MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000
UE-1R05	グラフィックボード AXシリーズ	92,000	55,000	CZ-8BE2	320K外部メモリー	29,800	25,300	MZ-1R29A	MZ-1P22増設RAM	32,000	12,000
UE-1R01	2M RAMボード AXシリーズ	300,000	特価	CZ-8BR1	立体映像セット	39,000	33,800	MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
UE-1R06	ROM ボード AXシリーズ	32,000	25,600	CZ-8BV2	カラーイメージボード	39,800	32,000	MZ-1T02	MZ-2200 テータレコーダー	19,800	8,500
UE-1R02	2M RAMボード AXシリーズ	100,000	特価	CZ-8BO1	FDインターフェイス	14,800	8,000	MZ-1T03	MZ-5500 テータレコーダー	12,000	8,500
UE-1U01	X 286L スロットBOX AXシリーズ	5,000	4,000	CZ-8BM2	モデムユニット	49,800	39,800	MZ-1U09	MZ-2500 拡張ボード	9,000	7,200
AX286D-FH4	MZ-8306A	458,000	特価	CZ-8EB3	拡張 I/Obox	33,800	28,000	MZ-1V01	パソコン FAX	278,000	85,300
AX286D-F	MZ-8302A	278,000	特価	CZ-8LM1	232cケーブル	7,200	6,000	MZ-1X22	モデムユニット	21,800	13,000
AX286L-F	MZ-8352A	428,000	特価	CZ-8LM2	232cクロスケーブル	7,200	6,000	MZ-2Z014	MZ-5500 TODAY	25,000	15,000
AX286L-FH3	MZ-8353A	598,000	特価	CZ-8NJ1	ジョイカード	1,700	1,360	MZ-2Z016	MZ-5500 附属		5,000
AX386-FH4	MZ-8706A	1,100,000	特価	CZ-8NT1	トラックボール	13,800	11,500	MZ-2Z028	MZ-6500 MSDOS.GWBASIC	60,000	35,000
AX386S-FH4	MZ-8706B	780,000	特価	CZ-8PK10	24ドット136桁漢字プリンター	99,800	69,000	MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
AX386-F	MZ-8702A	860,000	特価	CZ-8PK7	24ドット80桁漢字プリンター	22,000	59,800	MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
AX386S-F	MZ-8702B	590,000	特価	CZ-8PC4	24ドット熱転写カラー漢字プリンター	99,800	59,800	MZ-2Z029	MZ-6500 TODAY	68,000	20,000
AX386-FH8	MZ-8707A	1,280,000	特価	CZ-8TM1	モデムユニット300b	29,800	6,000	MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
AX386S-FH8	MZ-8707B	960,000	特価	AN-8TU	RGBシステムチューナー	33,100	特価	MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
CE-126P	ポケコンプリンター	17,800	13,600	AN-S100	アンプ付スピーカー	59,800	49,800	MZ-2Z012	MZ-5500 附属		5,000
CE-124	ポケコンカセットインター	4,500	3,600	HXD040	40Mハードディスク(1TM)	118,000	95,000	MZ-2Z013	MZ-5500 MS DOS	25,000	20,000
CE-120P	ポケコンプリンター	24,800	21,800	HXD140	40Mハードディスク内蔵用(1TM)	98,000	79,800	MZ-4Z001	MZ-5500IBM変換ユーティリティ	30,000	8,000
CE-123P	ポケコンプリンター	19,800	17,800	MZ-14FD	カラーディスプレイアダプタ0.31	49,800	特価	MZ-5511	本体	288,000	35,000
CE-140F	ポケコンフロッピーディスク	49,800	40,300	MZ-1D10	12"モノクロディスプレイ	41,800	25,000	MZ-5Z013	MZ-1500 QD通信ソフト		3,500
CE-140T	ポケコンRS-232Cコンバーター	9,800	8,800	MZ-1D17	15" CRT mZ-5500/6500/2	124,000	59,800	MZ-6BE2	X6800 2M RAM	35,000	23,500
CE-159	ポケコン RAM 8K	35,000	4,200	MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000	MZ-6F03	フランクQD DISK	450	400
CE-158	ポケコンレベルコンバーター	39,800	31,300	MZ-1E08	プリンター I/F 2000/2200/80B	9,000	8,000	MZ-6P18	MZ-1P18.28 カットシートフィーダー	60,000	35,000
CE-1600E	ポケコンディスプレイフェイス	19,800	17,800	MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000	MZ-6P11	MZ-1P10 カットシート	95,000	35,000
CE-1601M	ポケコン RAM 64K	45,000	30,000	MZ-1E04	MZ-2000プリンター I/F	10,000	6,000	MZ-6P29	MZ-1P28 カットシートフィーダー	50,000	37,500
CE-161	ポケコン RAM 16K	50,000	3,800	MZ-1E21	MZ-5500 GP I/F	36,000	12,000	MZ-6P27	MZ-1P27 カットシートフィーダー	58,000	39,800
CE-1650F	ポケコンDISK	9,800	8,800	MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000	MZ-6P06	MZ-1P06トラクターフィーダー	15,000	7,500
CE-1600P	ポケコンプリンター	69,800	59,800	MZ-1E33	MZ26500パラレル I/F	34,800	28,000	MZ-6P20	MZ-1P22/17ロールホルダー	3,100	2,700
CE-1600F	ポケコン フロッピードライブ	39,800	34,800	MZ-1E45	MZ-6500 232C I/F	50,000	15,000	MZ-6Z22	M-50 CP/M86	10,000	6,000
CE-1600M	ポケコン RAM 32K	32,000	16,000	MZ-1E32	MZ2500パラレル I/F	30,000	27,000	MZ-6Z25	M-50 ストリムーティライズアクセサ	39,800	15,000
CE-201M	ポケコン RAM 8K	18,000	3,000	MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000	MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
CE-202M	ポケコン RAM 16K	35,000	6,000	MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000	MZ-80TUB	MZ-80 バックアップ	20,000	8,000
CE-203M	ポケコン RAM 32K	32,000	7,000	MZ-1E29	RS-232Cインターフェイス	17,800	9,800	MZ-80P4B	136桁ドットプリンター		48,000
CE-T800	ポケコンRS-232Cコンバーター	12,800	11,800	MZ-1E01	MZ-3500 232Cボード	28,000	13,000	MZ-80TU	MZ-80 システムプログラム	20,000	8,000
CZ-300F	X13"マイクログロッピー	79,800	9,000	MZ-1E14	MZ1500QD用インターフェイス	9,800	3,000	MZ-80T40A	MZ-80 PASCAL	10,000	5,000
CZ-31F1	300F増設フロッピー	59,800	7,000	MZ-1M01	MZ-2000/220016ビットボード	78,000	8,000	MZ-80T70A	MZ-80 FDOS	20,000	7,000
CZ-501H	X1増設用ハードディスクユニット	258,000	特価	MZ-1M09	MZ-6500 8082-2演算プロセッサ	82,000	30,000	MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
CZ-503F	CZ-830増設ドライブ	49,800	30,000	MZ-1M03	MZ-5500 数値演算	69,000	38,500	MZ-8BIO4	MZ2000/2200 GP.IBインターフェイス	45,000	18,000
CZ-520F	2HD/2DDミニフロッピードライブ	118,000	70,000	MZ-1M12	MZ-2861 8087 演算プロセッサ	90,000	45,000	MZ-8BGRM1	MZ-80 BGRAM1	39,000	10,000
CZ-6BGI	GPIBボード	59,800	47,800	MZ-1P06	ドットプリンター	234,000	45,000	MZ-8BC01	MZ2000/2200 GP.IBケーブル	18,000	8,000
CZ-6BP1	数値演算ボード	79,800	63,800	MZ-1P28	ドットプリンター 漢字80桁	148,000	118,400	MZ-8BD02	MZ-80 BFDOS	50,000	18,000
CZ-6BC1	FAXボード	79,800	65,000	MZ-1P10A	24ドットプリンター 漢字80桁	245,000	79,000	PC-1280	ポケコン	24,800	19,600
CZ-6BU1	ユニバーサル I/Oボード	39,800	33,800	MZ-1P22	熱転写漢字プリンター	59,800	25,000	PC-1248DB	ポケコン	11,000	9,800
CZ-6BM1	MIDIボード	29,800	23,800	MZ-1P29	漢字プリンター 136桁	168,000	134,400	PC-1262	ポケコン	24,800	19,600
CZ-6BE1A 1M	X68000増設RAMボード	38,000	19,500	MZ-1P30	136桁プリンター	228,000	120,000	PC-1360	ポケコン	29,800	19,800
CZ-6BE1B 1M	X68000増設RAMボード		19,500	MZ-1R01	MZ-2000/2200Gボード	39,800	10,000	PC-1360K	ポケコン	36,800	32,800
CZ-6BE1 1M	X68000増設RAMボード	35,000	29,500	MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800	PC-1600K	ポケコン	69,800	49,800
CZ-6BN1	スキャナーボード	29,800	25,300	MZ-1R09	MZ-5500 V.RAM	35,000	15,000	PC-E500	ポケコン	28,800	19,800
CZ-6BE4	X68000増設RAMボード	138,000	特価	MZ-1R06	MZ-5500増設RAM	45,000	8,000	PC-E550	ポケコン	32,000	特価
CZ-6BF1	RS-232C 増設ボード	49,800	42,300	MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000				

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウェア全種取扱います。

X68000 全機種取り揃え大特価セール

新店舗(京王線・北野駅前)オープン記念セール実施中!



ALBIT

アイビット電子株式会社

京王線・北野駅前店

ミニ電腦遊園地/12月8日(土)・9日(日)開催

SHARP X68000シリーズ対応 ハードディスク

(ITEM)

HXD 040 X68000

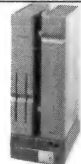
定価¥118,000→特価¥95,000

HXD 042 X68000 増設用

定価¥128,000→特価¥102,500

HXD 140 X68000 内蔵用

定価¥98,000→特価¥79,800



マイコン、パソコン、
ワープロ教室開催中!

SHARP X68000

※特価表示はTELにてご確認ください。

CZ-603C (本体)

プラス(ディスプレイ) 組合せ

CZ-602DBK	特価
CZ-603D	特価
CZ-611DGY	¥305,000
CZ-613D	特価

CZ-652C (本体)

プラス(ディスプレイ) 組合せ

CZ-602DBK	¥275,000
CZ-603DB	¥260,000
CZ-612DGY	¥290,000
CZ-605D	¥290,000

CZ-612CBK (本体)

プラス(ディスプレイ) 組合せ

CZ-603DBK	¥330,000
CZ-605DBK	¥360,000
CZ-613DBK	¥370,000
CZ-602DBK	¥345,000

CZ-663C (本体)

プラス(ディスプレイ) 組合せ

CZ-604D	特価
CZ-611DGY	¥385,000
CZ-612DGY	¥400,000
CZ-605D	特価

CZ-653C (本体)

プラス(ディスプレイ) 組合せ

CZ-602DBK	特価
CZ-603D	特価
CZ-612DGY	¥290,000
CZ-605D	特価

CZ-602C (本体)

プラス(ディスプレイ) 組合せ

CZ-603DGY	¥270,000
CZ-613DGY	¥310,000
CZ-605DGY	¥300,000
CZ-611DGY	¥285,000

CZ-613CBK (本体)

プラス(ディスプレイ) 組合せ

CZ-604D	¥410,000
CZ-605D	¥430,000
CZ-613D	¥440,000
CZ-21HD	¥450,000

CZ-623CTN (本体)

プラス(ディスプレイ) 組合せ

CZ-611DGY	¥445,000
CZ-612DGY	¥460,000
CZ-613DTN	特価
CZ-21HD	特価

アイビット推奨ディスプレイ

●シャープCZ-860D・BK
カラーディスプレイ
0.31ピッチ付A/D 15/24
定価¥92,200→
特価¥59,800



CZ-860D 対応パソコン機種: CZ880C/881C、X1/
TURBOシリーズ。ケーブルは本体付属を使用。
NEC PC-8801・9801シリーズ(XA・XLのみ不可)
MZ700/1500/2000/2200/2500各シリーズ(推奨
品シャープ8D8K)。(ドットピッチ0.39)

●シャープCZ-603D・GY・BK
(15型カラーディスプレイ)
ドットピッチ3.9
定価¥84,800→
特価



CZ-603D 対応パソコン機種: *X1シリーズ/*
X1 turboシリーズ/X1 turbo Zシリーズ/X68000
シリーズ/PC8801シリーズ/PC-9801シリーズ/
PC-286シリーズ
(※は接続ケーブルANI506が必要です)

●シャープCZ-830D・BK
(14型)
2モードオートスキャン方式
(アナログ/デジタル)
定価¥98,000→
特価¥54,800(在庫限り)



CZ-830D 対応パソコン機種: CZ880C/881C、X1/
TURBOシリーズ。ケーブルは本体付属を使用。
NEC PC-8801・9801シリーズ(XA・XLのみ不可)
MZ700/1500/2000/2200/2500各シリーズ(推奨
品シャープ8D8K)。

●シャープCZ-602D・BK
(15型アナログTV/3モード
オートスキャン)
定価¥99,800→
特価¥75,000



CZ-602D 対応パソコン機種: *X1シリーズ/*
X1 turboシリーズ/X1 turbo Zシリーズ/X68000
シリーズ/PC8801シリーズ/PC-9801シリーズ/
PC-286シリーズ
(※は接続ケーブルANI506が必要です)

●三菱XC-1498CII
(14型アナログ)
ドットピッチ0.28
定価¥107,000→
特価¥59,800



XC-1498CII 対応パソコン機種: PC-9801シリーズ/
PC-286シリーズ/PC-386シリーズ/PC-8801
シリーズ
(上記機種には付属の接続ケーブルで、接続可能)

※シャープ周辺機器(拡張、プリンター他)も常時取り扱っております。

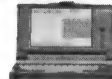
SHARP
AX286L-F
ラップトップ
定価¥428,000
→在庫処分中!



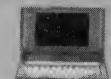
SHARP
AX286N-H2
All in Note
定価¥398,000
→大特価!



NEC
PC-9801N
NOTE
定価¥248,000
→特価¥198,000



TOSHIBA
J3100SS
Dyna Book
定価¥198,000
→特価¥149,000



富士通FM TOWNSお買得セット

FM TOWNS
モデル1基本セット

FM TOWNS-1	¥338,000
FMT-DP531	¥89,000
FMT-KB101	¥20,000
B276A010	¥20,000
特選ラック	¥24,000
定価合計	¥491,800
特価	¥198,000

FM TOWNS
モデル2基本セット

FM TOWNS-2	¥398,000
FMT-DP531	¥89,800
FMT-KB101	¥20,000
B276A010	¥20,000
特選ラック	¥24,000
定価合計	¥551,800
特価	¥278,000

FM TOWNS
モデル1拡張セット

FM TOWNS-1	¥338,000
HM-01T	¥32,800
FMT-DP531	¥89,800
FMT-KB101	¥20,000
B276A010	¥20,000
FMT-FD301	¥28,000
特選ラック	¥24,000
定価合計	¥552,600
特価	¥258,000

FM TOWNS
モデル1S基本セット

FM TOWNS-1S	¥338,000
FMT-DP531	¥89,800
FMT-KB101	¥20,000
B276A010	¥20,000
特選ラック	¥24,000
定価合計	¥491,800
特価	¥238,000

FM TOWNS
モデル1S拡張モデル2セット

FM TOWNS-1S	¥338,000
HM-01T	¥32,800
FMT-DP531	¥89,800
FMT-KB101	¥20,000
B276A010	¥20,000
特選ラック	¥24,000
定価合計	¥524,600
特価	¥268,000

FM TOWNS
モデル2Fセット

FM TOWNS-2F	¥378,000
FMT-KB101	¥20,000
FMT-DP531	¥89,800
B-276A010	¥20,000
特選ラック	¥24,000
定価合計	¥531,800
特価	¥310,000

(TOWNSお買い上げの方)パソコン教室が御利用できます。初・中・上級者 無料にて実施中!

〈全商品新品完全保証付〉■シャープポケコン全商品販売中。カタログ、特価表ご請求ください(〒72)

0426-45-3002(駅前店) 3001(本店)
FAX.0426-44-6002

●営業時間/10:00~19:00●電話受付/20:00迄可●定休日/水曜日

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●この広告の商品にはすべて送料・消費税は含まれておりません。

上記の広告商品はすべて店頭販売もしております

全通販
国信売

北海道から沖縄まで

富士銀行八王子支店 (普)1752505

OAB



オーエーブレイン

全国通販

☎03-5688-3621

OAB

●オフコンからパソコンまで
幅広～い品揃え。おまかせあれ!! お電話くださいネ!

- ★全商品保証書付。専門のアドバイザーがお客様のニーズに親切に対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。
★送料は1個につき¥1,000です。(※一部離島は除きます。お問合せ下さい。)

- ご注文、お問合せは…毎日午前10時から午後8時まで
●下取・買取は電話で見積りしております。責任を持って下取りさせていただきます。
●商品のお届けは…入金確認後、即日発送致します。

OAB特選～X68000シリーズセット (ゲームパック・ディスク付) (税抜き)

①X68000 EXPERTII

- CZ-603C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥453,000

クレジット例

1回	¥345,000	12回	¥30,200×12
----	----------	-----	------------



●SX-WINDOW搭載!!

②X68000 EXPERTII-HD

- CZ-613C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥563,000

クレジット例

1回	¥428,000	12回	¥37,500×12
----	----------	-----	------------

③X68000 PROII

- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥400,000

クレジット例

1回	¥297,000
12回	¥26,000×12

●SX-WINDOW搭載!!



④X68000 PROII-HD

- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥510,000

OAB大特価

X68000 SUPER-HD ●SX-WINDOW搭載!!

- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック

クレジット例

1回	¥485,000	12回	¥42,000×12
----	----------	-----	------------



OAB大特価

⑤X68000 SUPER-HD

- CZ-623C-TN(チタン)
- CZ-613D-TN(チタン)
- MD-2HD 20枚

定価合計 ¥633,000

OAB大特価

OAB大特価

OAB大特価

X68000 特選OABセット (中古美品)

①CZ-623C-TN+CZ-602D

2台限り

¥460,000

②CZ-662C(BK)+CZ-605D

5台限り

¥298,000

③CZ-602C(BK)+CZ-605D

2台限り

¥273,000



④CZ-881C(BK)

2台限り

¥47,000

⑤CZ-880D(BK)

3台限り

¥49,000

⑥CZ-830D(BK)

2台限り

¥41,000

周辺機器コーナー

プリンターセットコーナー

- CZ-6PVI(カラービデオプリンター)
定価 ¥198,000 ▶特価¥152,000
- CZ-8P03(24ドット熱転写カラープリンター)
定価 ¥65,800 ▶特価¥ 53,000
- CZ-8PK10(24ピン漢字ドットプリンター・136桁)
定価 ¥97,800 ▶特価¥ 73,000
- CZ-8PGI(24ピンカラー漢字ドットプリンター・80桁)
定価 ¥130,000 ▶特価¥ 98,000
- CZ-8PG2(24ピンカラー漢字ドットプリンター・136桁)
定価 ¥160,000 ▶特価¥119,000
- IO-735X(カラーイメージジェットプリンター)
定価 ¥248,000 ▶特価¥185,000

X68000用ソフトウェア・コーナー

- ①CZ-212BS(BUSINESS) ……定価 ¥68,000▶特価¥ 53,000
- ②CZ-220BS(DATA) ……定価 ¥58,000▶特価¥ 45,000
- ③CZ-215MS(Sampling) ……定価 ¥17,800▶特価¥ 13,800
- ④CZ-221HS(NEW Print Shop) ……定価 ¥10,800▶特価¥ 15,500
- ⑤CZ-227BS(TOP財務会計) ……定価 ¥200,000▶特価¥158,000
- ⑥CZ-226BS(GARD) ……定価 ¥229,800▶特価¥ 23,000
- ⑦CZ-223CS(Communication) ……定価 ¥19,800▶特価¥115,500
- ⑧CZ-213MS(MUSIC) ……定価 ¥18,800▶特価¥ 14,800
- ⑨CZ-211LS(C compiler) ……定価 ¥39,800▶特価¥ 31,000
- ⑩C-TRACE(キャス) ……定価 ¥68,000▶特価¥ 52,000
- ⑪EW(イースト) ……定価 ¥38,000▶特価¥ 29,000

X68000用周辺機器コーナー

- CZ-6BE1B ……定価 ¥28,000▶特価¥ 22,000
- CZ-6BM1 ……定価 ¥26,800▶特価¥ 21,000
- CZ-6EB1 ……定価 ¥88,000▶特価¥ 69,800
- CZ-6VT1 ……定価 ¥69,800▶TEL下さい
- CZ-6NS1 ……定価 ¥188,000▶特価¥149,000
- CZ-6BC1 ……定価 ¥79,800▶特価¥ 63,000

■CZ-8PC4(定価 ¥99,800)

特選品!!

- 48ドット熱転写カラー漢字プリンター

特価¥64,800



今月の特価品 (限定)お早目に!!

★CZ-653C(BK)+CZ-602D(BK)

4セット限り ……大特価¥258,000

- SHARP WD-A320(ワープロ)
定価 ¥165,000 ……特価¥129,000
- SHARP WD-A341(ワープロ)
定価 ¥185,000 ……特価¥139,000
- SHARP WD+HL30(ワープロ)
定価 ¥198,000 ……特価¥120,000

●SHARP PW-910(ワープロ)

●NEC PC-KD853(アナログCRT) ……特価 ¥ 60,000

●三菱XC-I498C(アナログCRT) ……特価 ¥ 50,000

●SHARP CU-I4FD(アナログCRT) ……特価 ¥ 54,800

●SHARP PA-8500(電子手帳) ……特価 ¥ 46,000

●SHARP PA-8500(電子手帳) ……特価 ¥ 16,000

通信販売によるご購入方法(お電話でお申し込み下さい。)

現金一括払い	クレジット	振込先
銀行振込: 電話扱いにてお振込下さい 手数料はお客様負担となります	専用のお申し込み用紙をお送り致します のて、必要事項をご記入・捺印のうえ、ご返送下さい	●第一勧業銀行 御徒町支店 (普)1376679 オーエーブレイン ●朝日信用金庫 本店 (普)348433 オーエーブレイン
現金書留: 住所、氏名、電話番号、商品名、使用機種、 メタ印等をお書き添えのうえ、現金書留に 当封でお送り下さい	※未成年者の方は、保護者のご承認を 受けてからお申し込み下さい	
★クレジットは1～60回払いで月々5,000円より自由に設定できます。		

オーエーブレイン 〒110 東京都台東区台東1-28-4
TEL & FAX 5688-3621

I-O DATA 増設RAMボード

●1MB増設PAMボード
PIO-6BE1-A定価
¥25,000●2MB増設RAMボード
PIO-6BE2-2M定価
¥50,000●4MB増設RAMボード
PIO-6BE4-4M定価
¥88,000

特価¥18,500 特価¥37,500 特価¥65,000

■ハードディスク ■特価品もありますのでTEL下さい。

- アイテック ITX-640 ……特価 ¥117,000
- アイテック ITX-680 ……特価 ¥149,000
- ロジテック LHD-32V ……特価 ¥ 85,000
- ロジテック LHD-34VE ……特価 ¥ 90,000
- ロジテック LHD-34V ……特価 ¥104,000
- シャープ CZ-620H ……特価 ¥118,000
- シャープ CZ-64H ……特価 ¥ 95,000
- アイテム HXD-040 ……特価 ¥ 88,000
- アイテム HXD-042 ……特価 ¥ 95,000
- ICM SR 80 ……特価 ¥130,000

中古パソコン (価格/在庫は変動します。予約は5日以内とします。)

- PC-9801RA5 ……¥338,000より
- PC-9801RA2 ……¥265,000より
- PC-9801RX2 ……¥199,000より
- PC-9801EX2 ……¥190,000より
- PC-9801VX2 ……¥170,000より
- PC-9801UX2 ……¥165,000より
- PC-9801VX2 ……¥160,000より
- PC-9801VX2 ……¥150,000より
- PC-9801UV1 ……¥148,000より
- PC-9801LV2 ……¥160,000より
- PC-286VE ……¥150,000より
- PC-286US ……¥155,000より
- PC-286VS ……¥165,000より
- CZ-600C ……¥145,000より
- CZ-601C ……¥158,000より
- CZ-611C ……¥168,000より
- CZ-652C ……¥148,000より
- CZ-612C ……¥210,000より
- 68000用モニター ……¥ 49,000より
- PC-9801用サウンドボード ……¥ 13,000より
- PC-88SR, FR ……¥ 50,000より
- PC-88FH, FA ……¥ 65,000より
- 400ラインCRT ……¥ 38,000より
- 200ラインCRT ……¥ 10,000より

オーエーブレイン今月の特価品!! 台数限定 お早目に!!

ドライブ・ユニット	プリンター	ハード・ディスク
アクセラ ●FD-357 ……特価¥36,000 ●FD-358 ……特価¥49,000 コンピュータ・リサーチ ●CRC-FD3.5S ……特価¥29,000 ●CRC-FD3.5W ……特価¥42,000 グローリア ●GD-35M1 ……特価¥23,000 ●GD-35M2 ……特価¥39,000 緑電子 ●Little-F1 ……特価¥26,000 ●Little-F2 ……特価¥38,000 SNE ●SNE-2 ……特価¥49,000	NEC ●PC-PR201G+ ……特価 ¥99,800 ●NM-4150 ……特価 ¥132,000 1 サル ●AP-850PC ……特価 ¥ 64,000 ●VP-2050PC ……特価 ¥ 99,000 ●BJ-130J ……特価 ¥125,000 SNE 1 サウンドオーストラ ……特価 ¥23,000 2 サウンドオーストラ ……特価 ¥17,800 3 サウンドオーストラ ……特価 ¥14,500 4 サウンドオーストラ ……特価 ¥12,000 5 サウンドミュージシャン ……特価 ¥16,000 6 サウンドミュージシャン ……特価 ¥ 8,500	ARK WOOD NEC純正ドライブ使用 ●AW-N40C ……(定価 ¥138,000) 特価 ¥ 94,000 ●AW-N100C ……(定価 ¥195,000) 特価 ¥134,000 ワープロ ●PWP-70HR ……特価 ¥175,000 ●PWP-70R ……特価 ¥124,000 ●PWP-50RD ……特価 ¥119,000 ●PWP-50RD ……特価 ¥140,000 ●JW-95FD ……特価 ¥108,000 ●CW-350 ……特価 ¥155,000 その他、シャープ、松下、富士通等、TEL下さい

■流通事情により、広告表示よりお安くなる場合もございます。まずは、お電話下さい。■ビジネス・ゲームセットもございます。

全国通販

高中価買取中!

注文書				お名前	買取依頼書				
ソフト名	機種	メディア	販売価格	住所	ソフト名	機種	メディア	販売価格	
<div style="position: absolute; top: 50px; left: 50px; font-size: 2em; opacity: 0.5;"> 同人ソフト募集中! 「自宅で学ぶアセンブラ」 通信教育受付中 </div>					TEL				

新品ソフト15%OFF 送料、消費税込み。 ただし、北海道・沖縄・離島の方は200円プラスして送金して下さい。
定価5,000円未満の商品についてはプラス300円。

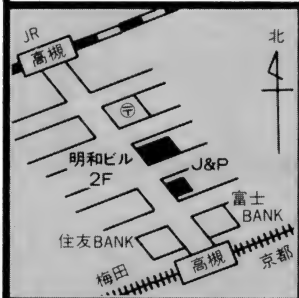
PC98シリーズ			X68000シリーズ		
商品名	定価(円)	販売価格	商品名	定価(円)	販売価格
三国志II	14,800	12,500	栄光は君に	9,500	8,000
ボビュラス	9,800	8,300	天下統一	9,800	8,300
MISTY6	5,000	4,200	熱血高校ドッジボール部サッカー編	8,800	7,400
ダンジョンマスター	9,800	8,300	メネシス'90	8,800	7,400
シムシティ	9,800	8,300			
大航海時代	9,800	8,300			
続ダンジョンマスターカオスの逆襲	9,800	8,300			
バーズテイル	9,800	8,300			
キサナ	6,800	5,700			
ドラゴンナイトII	6,800	5,700			
機動戦士ガンダム デザート	9,800	8,300			
サークII	8,800	7,400			
D.P.S.SG	6,800	5,700			
満開電飾	7,800	6,600			
ヒルズファー	9,800	8,300			
F15スライクイーグルII	10,800	9,100			
46億年物語	9,800	8,300			
煩悩予備校	7,800	6,600			
イルミナ	6,800	5,700			
ダークレイスII	7,800	6,600			
デジャブI	8,800	7,400			
ナビチュールドラゴン航海記	8,800	7,400			

中古ソフト

中古リストご希望の方は62円切手3枚をお送り下さい。

PC98シリーズ			X68000シリーズ		
商品名	定価(円)	販売価格	商品名	定価(円)	販売価格
三国志II	14,800	12,500	アフターバーナー	9,200	
ボビュラス	9,800	8,300	イース3	8,700	
ダンジョンマスター	9,800	8,300	AXIS	8,800	
サイレントメビウス	14,800	12,500	信長戦国群雄伝	9,800	
キャンベーン版大戦略2	9,800	8,300	シムシティ	9,800	
ドンネルズ&トロールズ	9,800	8,300	グラナダ	8,800	
FOXY	6,800	5,700	エージャックス	8,800	
ドラゴンナイト	6,800	5,700	ジェノサイド	8,800	
栄冠は君に	9,500	8,000	ナイトアームス	9,700	
インベリアルフォース	8,800	7,400	サラマンダー	8,800	
ダークレイス	9,600	8,100	スーパーハンクオン	8,800	
エイトレイクスゴルフクラブ	4,800	4,000	天下統一	9,800	
シムシティ	9,800	8,300	ダンジョンマスター	9,800	
プリンスオブペルシャ	8,800	7,400	ボビュラス	9,800	
ドラゴンズレイヤーVI	8,700	7,300	デスプリンガー	12,800	
維新の嵐	9,800	8,300	大海令	9,800	
提督の決断	14,800	12,500	ラストハルマゲドン	9,800	
水滸伝	9,800	8,300	三国志II	14,800	
バトル	12,800	10,800	メタルサイト	8,800	
ワンダラーズフロムイース	8,700	7,300	V'BALL	7,900	

- 代金は注文書添えて、現金書留で送って下さい。(小為替不可) 後払いシステムもあります。
- 新品ソフトをご注文の場合は、商品代金を送って下さい。(送料、消費税込み)
- 中古ソフトをご注文の場合は、必ず電話にて在庫確認をして下さい。
- 未発売ソフトの場合は、予約扱いとさせていただきます。
- 買取希望の場合は、まずソフトを当店に送って下さい。こちらで高額査定のため、TELでご連絡させていただきます。値段が合わない場合、商品はすぐ返送しますので、安心して送り下さい。
- ディスクの送料は、100枚まで500円です。
- DISKシャトル フランチャイズ店募集開始。



DISKシャトル

営業時間 AM12:00~PM8:00

高槻店

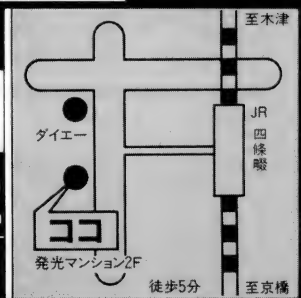
〒569 大阪府高槻市高槻町12-13
明和ビル2F

四條畷店

〒574 大阪府大東市北楠里27-19
発光マンション202

☎0726-83-9907 ☎0720-78-1342

東京地区TELオープン☎03-713-1424



SHARP

X68000

X68000

X68000

クリエイイト特典

- 全商品完全保証書付(メーカー保証)
- 全国無料配達(一部離島の方は有料になります)
- 配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- 中古パソコン高価下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00~PM7:00

当社はX68000の販売認定店です。
どんなことでも安心してご相談ください。

秋のX68000フェア実施中! 即売・即納

X68000 NEW PRO II

- CZ-653C(本体).....¥285,000
- CZ-603D(カラーディスプレイ).....¥84,800
- 好きなゲームソフト1本.....¥7,800
- 定価合計.....¥377,600

クリエイイト特価

均等払い	¥7,680×48回	¥9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 EXPERT セット

5台限定!

- CZ-602C-GY(本体).....¥356,000
- CZ-603D-GY(カラーディスプレイ).....¥84,800
- 定価合計.....¥440,800▶大特価¥279,000

大特価¥279,000

均等払い	¥12,850×24回	¥8,870×36回	¥6,920×48回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- CZ-603C(本体).....¥338,000
- CZ-613D(カラーディスプレイテレビ).....¥135,000
- CZ-8NJ2.....¥23,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥506,600

クリエイイト特価

均等払い	¥9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 SUPER HD

- CZ-623C-TN(本体・キーボード・マウス).....¥498,000
- CZ-613D-TN(カラーディスプレイ).....¥135,000
- CZ-6BP1.....¥79,800
- 定価合計.....¥712,800

クリエイイト特価

均等払い	¥7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000×8回	¥50,000×6回	¥80,000×4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ~!

- CZ-603C.....¥338,000
- CZ-605D.....¥115,000
- MU1.B(MIDIボード&ソフト).....¥39,800
- CM32L.....¥69,000
- グラナダ.....¥8,800
- JOYカード.....¥1,800
- 定価合計.....¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET!

- PRO II CZ653C.....¥285,000
- Q.31CRT CZ603D.....¥84,800
- グラナダ.....¥8,800
- Y'S.....¥8,700
- ポピュラス.....¥9,800
- スーパーハンガオン.....¥8,800
- エージャックス.....¥8,800
- サーク.....¥8,800
- アールタイプ.....¥7,800
- アナログJOYSTICK XE-1AP.....¥13,800
- 定価合計.....¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥69,800	MUSIC PRO	MIDI版	¥28,800
CZ-8NS1	カラーイメージキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥18,800
CZ-6BE1A	1MB増設RAMボード	¥38,000	SOUND PRO-68K	サウンドエディタ	¥15,800
CZ-6BE2	2MB増設RAMボード	¥79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥28,800
CZ-8NM3	マウス・トラックボール	¥9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥29,800
BF-68PRO	高性能CRTフィルター	¥19,800	PRO-68K	サイバーノート	¥19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥79,800	PRO-68K	ステーションナリー	¥14,800
CZ-8NT1	トラックボール	¥13,800	Ccompiler PRO-68K	ソフト開発セット	¥39,800
CZ-6BM1	MIDIボード	¥26,800	Human 68K Ver.2.0	開発ツールセット	¥9,800
CZ-8NJ2	アナログスティック	¥23,800	PIO-6BE1-A	内蔵1MRAM	¥25,000
CZ-6TU	パソコンチューナ	¥33,100	PIO-6BE2-2M	2MRAM	¥50,000
SX-68M	MIDI I/F	¥19,800	PIO-6BE4-4M	4MRAM	¥88,000
XE-1AP	アナログジョypad	¥13,800	MU1-B	MIDI I/F+ソフト	¥39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際にお問合せください。●超特価販売中!

オール15%~20%OFF

パソコン専門ショップ

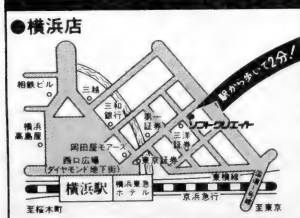
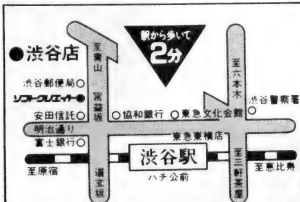
総合お問合せ先 ☎03-486-6541代

ソフトクリエイイト 渋谷/横浜

●渋谷店 ☎03-486-6541(代) 〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店①No.5000340

●横浜店 ☎045-314-4777(代) 〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店②No.310852

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。



Mu-1

Musicstudio [ミュージーワン]

Ver.1.4

Mu-1バージョンアップのお知らせ

Mu-1がVer.1.4にバージョンアップしました。

登録ユーザーの方に無料バージョンアップサービスを実施しております。

まだ登録されていない方は、早めにユーザー登録葉書をお送りください。

◆好評発売中◆

Mu-1 Ver.1.4 ¥19,800

Mu-1B Ver.1.4 ¥39,800

●システムサコム社製MIDIボード(SX-68M)付
●オリジナルI/Oスロットカバー同梱



Ver.1.4の特長

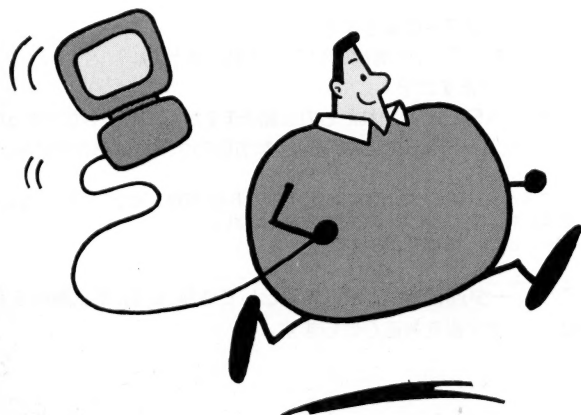
1. ミュージックコンバーターVer.2.0搭載
* ピッチベンド幅、MIDIチャンネル、トランスポーズ、リバーブデータ、等エクスクルーシブデータをMu-1フォーマットにコンバートします。
2. 内蔵FM音源対応
* Mu-1トラックデータで内蔵FM音源を鳴らすことができます。1トラック単音発音(和音の場合は、後着データが優先します。同時発音8音)。また、CM-32L、CM-32P風FM音色データを付録しました。
3. グラフィックチェインプレー機能
* 曲中でグラフィックデータを交換することができます。さらにイメージユニットの取り込み画像をセーブすることができます。
4. その他



SAN MUSICAL SERVICE
〒154 東京都世田谷区池尻4-1-4 ☎(03)419-8839

いつでも、どこでも ソフトバンクの11大雑誌

SOFT
BANK

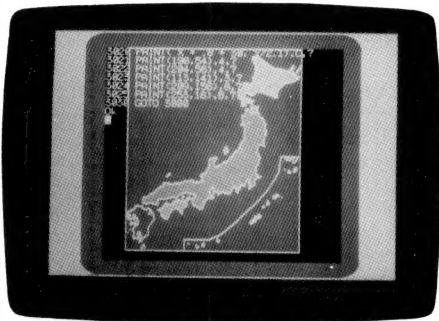


The SUPER FAMICOM
11 WEEK
THE COMPUTER
MAGAZINE
Oh! FM
Oh! PC
MEGADrive
Oh! Dyna
パソコンマガジン
月刊 情報処理試験

エミュレータ

好評発売中

定価¥9,800



X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3~5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。
この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。
このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリーイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

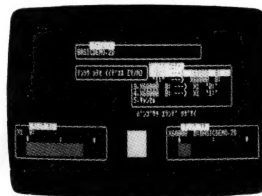
X1ディスク ↔ X68000 Human68k (5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC: CP/M ↔ X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。
- ※ 付属の専用ケーブルをX1とX68000に接続してファイルを転送します。



エミュレータ Q&A

- Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
A. 専用のケーブルが付属しますのでその必要はありません。
- Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。
- Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。
- Q. Turbo用のソフトは動きますか?
A. X1用のみでTurbo専用のソフトは動きません。
- Q. ゲームは動きますか?
A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。
- ※ タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
※ 一部サポートしていない機能があります。
- X1エミュレータ通信販売** 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

* この商品価格には消費税は含まれておりません。
* CP/Mはデジタルリサーチ社の商標です。
文中のソフトウェアは各社の商標です。
* 製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社 **アクセス** 〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
TEL 03 (233) 0200 (代) FAX 03 (291) 7019

J&P HOT LINE

SIG探訪PART 6

'88スキーツアー

▼(志賀高原・一ノ瀬スキー場)

スキー大好き!

(ジャンプコード:SKI)

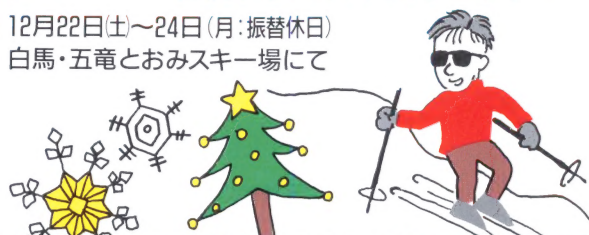


参加者募集中!

恒例クリスマス・スキーツアー

12月22日(土)~24日(月:振替休日)

白馬・五竜とおみスキー場にて



こんな人ぜひおいで!

とにかくスキーが大好きで、
1年中雪を夢見ている人、
スキーをしてみたいけど初めてだから不安な人、
忙しくてなかなかスキーに行けないから、
せめてスキーの話をして楽しみたい人、
スキーの趣味を共通点にして
仲間の輪を広げたい人
etc.

'89 クリスマス・スキーツアー
(志賀高原 熊の湯スキー場)

思い出はシュプールにして心の中に……。だから1年中がスキーシーズン!!

スキーのSIGといえばもちろん、スキーツアー! 金曜日の夜に出発して月曜日早朝に戻ってくる……こんなツアーをシーズン中に数回行っています。まるで修学旅行か合宿のようにワイワイ、ガヤガヤとお祭り気分。そのメインとなるのが恒例の“クリスマス・スキーツアー”。**自分の滑りを初めて見れる! と好評のビデオ撮影会**やクリスマスパーティー、プレゼント交換会等、もりだくさんの企画で楽しくやっています。もちろん**初心者にはスキースクールを実施**。スキーテクニックはもちろん、女性とペアリフトに乗る技術(?)まで親身にお教えしています。スキーツアーを行うのが前提のSIGなので、オフシーズンでもオフラインミーティングを大切にしており、**ニューモデル発表会の見学や人工スキー、バーベキューetc.**、こまめに集まって親睦を深めています。

その他 楽しいメニューがまだまだいっぱい!

- ★J&Pならではのパソコン・家電製品
の会員割引もある**ONLINE SHOPPING**。
- ★J&Pだから強い!! パソコン情報をはじめとする
役に立つ**DATA BASE**。
- ★みんなでおしゃべり**オンライントーク**(CHAT機能)。
- ★地域別・テーマ別ボードで充実の**BBS**(電子掲示板)。
- ★ビジュアルデータもばっちり送受信できる**X-MODEM**。

J&P HOT LINEへのご入会はスタータキットで。

買ったその日から
2週間無料で
アクセスできます。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店 東京都渋谷区道玄坂2丁目28番4号 ☎(03) 496-4141
町田店 東京都町田市森野1丁目39番16号 ☎(0427) 23-1313
八王子店 東京都八王子市旭町1番1号八王子そごう ☎(0426) 26-4141
立川店 東京都立川市幸町4-39-1 ☎(0425) 36-4141
本厚木店 厚木市中町3-4-3 ☎(0462) 25-1548
富山店 富山市桜町2-1-10 ☎(0764) 32-3133
金沢店 金沢市入江2-63 ☎(0762) 91-1130
金沢店 金沢市寺地2-3 ☎(0762) 47-2524
大須店 名古屋市中区大須4丁目2-48 ☎(052) 262-1141

新テコランド
メディアランド
コスモランド
U. S. LAND
ビジネスランド
梅田店 大阪市北区小松原町1-10 ☎(06) 362-1141
高槻店 高槻市高槻町11番16号 ☎(0726) 85-1212
くすは店 枚方市楠葉花園町15番2号 ☎(0720) 56-8181
千里中央店 豊中市新千里東町1-3 SENCHU PAL 2番街 ☎(06) 834-4141
摂津富田店 摂津市大畑町24-10 ☎(0726) 93-7521
寝屋川店 寝屋川市緑町4-20 ☎(0720) 34-1166

藤井寺店 藤井寺市岡2丁目1番33号 ☎(0729) 38-2111
岸和田店 岸和田市土生町2451-3 ☎(0724) 37-1021
さんのみやばい屋 神戸市中央区八幡通3-2-16 ☎(078) 231-2111
西宮店 兵庫県西宮市河原町5-11 ☎(0798) 71-1171
姫路店 姫路市東延町1丁目1番住友生命姫路ビルF ☎(0792) 22-1221
京都寺町店 京都市下京区寺町通仏光寺下ル徳美漬町54 ☎(075) 341-3571
京都近鉄店 京都市下京区烏丸通七条下ル東宝小路702 ☎(075) 341-5769
和歌山店 和歌山市元寺町4丁目4番地 ☎(0734) 28-1441
奈良1ばい館 奈良市三条町478-1 ☎(0742) 27-1111
郡山インテア 大和郡山市横田693-1 ☎(07435) 9-2221
熊本店 熊本市手取本町4-12 ☎(096) 359-7800

SHARP

ADVANCED TURBO

先駆の“Z”アビリティがパソコンクリエイターを魅了する。



AV パソコンテレビ turbo Z III

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK 標準価格 169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK 標準価格 92,200円(税別)
チルトスタンド	CZ-6ST1-B 標準価格 5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロップ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときには便利なグラフィック用関数(HSV、RGB、HALF、CDOWN、CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート ●1Mバイトの5インチフロッピーディスクドライブ2基搭載 ●JIS第1/第2水準漢字、システム・ユーザー辞書を標準装備した高度な日本語処理機能 ●ニューデザインのマウス標準装備 ●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計 ●プリンタ、RS-232Cなど豊富なインターフェイスを装備 ●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)

シャープ株式会社

●お問い合わせは…シャープ株式会社電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

T4910217912569 雑誌 02179-12